

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平9-179734

(43)公開日 平成9年(1997)7月11日

(51)Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/06	5 5 0		G 0 6 F 9/06	5 5 0 L
12/14	3 2 0		12/14	3 2 0 F

審査請求 未請求 請求項の数7 O L (全 16 頁)

(21)出願番号 特願平7-338392

(22)出願日 平成7年(1995)12月26日

(71)出願人 000000295

沖電気工業株式会社

東京都港区虎ノ門1丁目7番12号

(72)発明者 森田 幸伯

東京都港区虎ノ門1丁目7番12号 沖電気
工業株式会社内

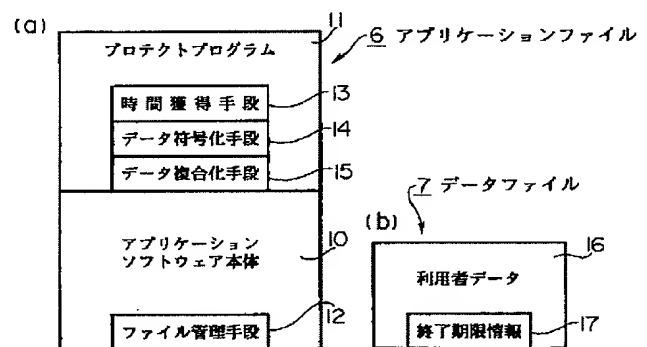
(74)代理人 弁理士 船橋 國則

(54)【発明の名称】 評価用ソフトウェアの不正使用防止方法

(57)【要約】

【課題】 データの継続的な不正使用を防止することにより、評価用ソフトウェアの継続的な不正使用を防止した、評価用ソフトウェアの不正使用防止方法の提供が望まれている。

【解決手段】 データファイルの使用を必要とする評価用ソフトウェアの不正使用防止方法である。データファイル7に使用期限情報を付加してこれに格納する処理と、評価用ソフトウェアを使用するプロセッサから日付情報を読み取る処理と、データファイル7に格納された使用期限情報とプロセッサから読み取った日付情報とを比較し、日付情報から得られる日付が使用期限情報で規定された使用期限を過ぎている場合に、データファイル7の使用を制限する処理と、を備えている。



実施形態例の説明図

【特許請求の範囲】

【請求項 1】 データファイルの使用を必要とする評価用ソフトウェアの不正使用防止方法であって、上記データファイルに使用期限情報を付加してこれに格納する処理と、

該評価用ソフトウェアを使用するプロセッサから日付情報を読み取る処理と、

上記データファイルに格納された使用期限情報と上記プロセッサから読み取った日付情報とを比較し、該日付情報から得られる日付が使用期限情報で規定された使用期限を過ぎている場合に、上記データファイルの使用を制限する処理と、を備えてなることを特徴とする評価用ソフトウェアの不正使用防止方法。

【請求項 2】 上記データファイルに使用期限情報を付加してこれに格納するに際して、該使用期限情報を暗号化することを特徴とする請求項 1 記載の評価用ソフトウェアの不正使用防止方法。

【請求項 3】 上記データファイルが複数あり、使用期限情報を付加していない新規のデータファイルに使用期限情報を付加するに際して、既に使用期限情報が付加されているデータファイルがある場合に、該使用期限情報によって規定された使用期限が最も早いものと同じ使用期限となるように使用期限情報を付加し格納することを特徴とする請求項 1 記載の評価用ソフトウェアの不正使用防止方法。

【請求項 4】 上記のデータファイルの使用を制限する処理が、評価用ソフトウェアのプログラムファイルおよびデータファイルを消去することにより、該評価用ソフトウェアの使用を不能にする処理であることを特徴とする請求項 1 記載の評価用ソフトウェアの不正使用防止方法。

【請求項 5】 データファイルの使用を必要とする評価用ソフトウェアの不正使用防止方法であって、上記データファイルに格納するデータの少なくとも一部を、該データの作成日に依存したキーに基づいて符号化し、その後この符号化したデータを上記データファイルに格納するとともに、この格納したデータについての複合化が正しく行えるか否かの検証を可能にした検証パターンを上記データファイルに格納する処理と、該評価用ソフトウェアの使用に際して上記データファイルのデータを使用する場合に、該評価用ソフトウェアを使用するプロセッサから日付情報を読み取る処理と、該日付情報を読み取った後、読み取った日付情報に依存したキーに基づいて上記検証パターンの複合化を行い、上記データファイルのデータについての複合化が正しく行えるか否かを検証する処理と、上記データファイルのデータについての複合化が正しく行えないと検証された場合に、上記データファイルの使用を制限する処理と、を備えてなることを特徴とする評価用ソフトウェアの不正使用防止方法。

【請求項 6】 上記データファイルのデータについての複合化が正しく行えるか否かを検証する処理が、該評価用ソフトウェアの使用に際して上記データファイルのデータを使用する場合に、上記プロセッサから読み取った日付情報に依存したキーに加え、一回、もしくは定められた回数の期間を遡った日付情報に依存したキーを用意し、これらキーに基づいて上記検証パターンの複合化を行う処理であることを特徴とする請求項 5 記載の評価用ソフトウェアの不正使用防止方法。

【請求項 7】 上記の複数のキーに基づいて上記検証パターンの複合化を行う処理が、上記データファイルに期間種別としてキー毎の情報を保持させることにより、各キー毎に複合化を行うことなくキー毎の情報に基づいて複合化を一回の処理で行うことを特徴とする請求項 6 記載の評価用ソフトウェアの不正使用防止方法。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】 本発明は、評価用ソフトウェアの不正使用防止方法に係り、詳しくはデータファイルの使用を必要とする評価用ソフトウェアの不正使用防止方法に関する。

【0002】

【従来の技術】 一般にソフトウェアの流通においては、ソフトウェアの正当な評価を広く行ってもらうため、評価版（評価用ソフトウェア）を配布することが行われる。このような評価版の配布は、あくまでソフトウェアの評価が目的であり、無償で貸与するのが普通であることから、日常業務などの実用のため、不正に用いられないようにすることが必要となっている。一般的に、このような不正使用の防止のためには、通常の機能のうち一部の機能、例えば印刷機能やファイル格納機能などを制限することが多くなされている。

【0003】 ところが、例えばデータベースなどのデータの格納が重要であるソフトウェアの場合では、特にデータの格納を制限してしまうことは望ましくなく、したがってファイル格納機能の制限を外して欲しいとの要望が多い。このような要望に応え、しかも不正使用の防止を図った技術として、従来、評価用ソフトウェアの使用期間や使用回数を限定する方法（特開平 4-54529 号公報）、さらにはパスワードを用い、このパスワードを判定することによって一定時間が経過すると不正使用できなくなるようにした方法（特開平 4-65716 号公報）が知られている。

【0004】 図 11 は、上記の評価用ソフトウェアの使用期間や使用回数を限定する方法において用いられる、不正使用防止が施されたプログラムファイルの構成の一例を示す図である。図 11 において符号 1 は評価用アプリケーションソフトウェア本体であり、2 はプロテクトプログラムである。プロテクトプログラム 2 には、評価終了期限に関する情報である終了期限情報 3 と、使用回

数情報 4 と、時間取得手段 5 とが備えられている。終了期限情報 3 は、評価開始時に開始時間を基に計算され、記録されてファイル中に保持されたものである。使用回数情報 4 は、評価用ソフトウェアが使用されるたびに更新されるもので、予め設定され記録された許容回数に比較されるものである。時間取得手段 5 は、プログラムが起動したときの開始時間を取得するもので、通常、オペレーティングシステムの時間獲得手段を介してハードウェアの時計機構などから時間情報を得るものである。そして、このような構成のもとにプロテクトプログラム 2 は、該評価用ソフトウェアの起動時に、起動時間と使用期限の比較、あるいは使用回数が予め設定された許容回数内であるか否かの判定を行い、許容範囲を越えている場合に、該評価用ソフトウェアの起動を中止し、もしくはソフトウェアファイルの消去を行うようになっている。

【0005】

【発明が解決しようとする課題】ところで、図 11 に示したプログラムファイルを用いる方法を採用する場合では、通常評価用ソフトウェアの設定（インストール）については該ソフトウェアを配布するメーカーが行うようになっている。しかしながら、近年では評価用ソフトウェアをより多くの利用者に評価してもらうため、ネットワークや CD-ROM による配布もなされるようになってきている。しかし、このようにネットワークや CD-ROM によって配布した場合、開始時間の設定をインストール時に行うと、この開始時間の設定が利用者の手に委ねられることになり、最初の終了期限の設定時に不正が行われるおそれが生じてしまう。例えば、複数回のインストールを行うことにより、長期に使用することが可能になってしまう。すなわち、ソフトウェアの設定前にプログラムファイルの複製を作成し、一定期間もしくは一定回数を評価した後データを保存し、別の評価として保存されたプログラムファイルより再設定して評価を開始し、保存されたデータを開くことにより、継続して該ソフトウェアを使用することが可能になってしまうのである。また、ソフトウェアと同時にデータファイルを消去する方法も考えられるが、データファイルは一般に複製可能であり、複製したものをを用いて再評価することにより、やはり該ソフトウェアを継続して使用することが可能になってしまう。

【0006】このような不都合を回避するため、前述したようにパスワードを用い、このパスワードを判定することによって一定時間が経過すると不正使用できなくなるようにした方法も提案されている。しかし、その場合には期限の設定を利用者に委ねないよう、同一期間のみ動作するように予め期限を一律に設定する必要がある。ところが、このように期限を一律に設定してしまうと、評価用ソフトウェアの入手が遅れた利用者にとっては、その使用期間が短くなってしまい、評価が十分に行えなく

なってしまうといった新たな不都合を生じてしまう。

【0007】本発明は上記事情に鑑みてなされたもので、データの継続的な不正使用を防止することにより、評価用ソフトウェアの継続的な不正使用を防止した、評価用ソフトウェアの不正使用防止方法を提供することを目的とするものである。

【0008】

【課題を解決するための手段】本発明における請求項 1 記載の評価用ソフトウェアの不正使用防止方法では、データファイルの使用を必要とする評価用ソフトウェアの不正使用防止方法において、上記データファイルに使用期限情報を付加してこれに格納する処理と、該評価用ソフトウェアを使用するプロセッサから日付情報を読み取る処理と、上記データファイルに格納された使用期限情報と上記プロセッサから読み取った日付情報とを比較し、該日付情報から得られる日付が使用期限情報で規定された使用期限を過ぎている場合に、上記データファイルの使用を制限する処理とを備えてなることを上記課題の解決手段とした。

【0009】この発明によれば、利用者の作成するデータファイルに使用期限情報を付加し、さらにこの使用期限情報とプロセッサから読み取った日付情報とを比較し、使用期限を過ぎている場合に上記データファイルの使用を制限する処理を行うようにしたので、上記データファイルの継続的な不正使用を防止することが可能になる。ここで、上記データファイルとしては、ワープロ等のデータファイルのみでなく、ソフトを使用する際の利用者の設定情報を格納するファイルも含む。なお、データファイルを作成することが必須でないソフトも多数あるが、このようなファイルをソフト起動と同時に作成することにより、本発明は全ての評価用ソフトウェアを対象とすることができる。

【0010】本発明における請求項 5 記載の評価用ソフトウェアの不正使用防止方法では、データファイルの使用を必要とする評価用ソフトウェアの不正使用防止方法において、上記データファイルに格納するデータの少なくとも一部を、該データの作成日に依存したキーに基づいて符号化し、その後この符号化したデータを上記データファイルに格納するとともに、この格納したデータについての複合化が正しく行えるか否かの検証を可能にした検証パターンを上記データファイルに格納する処理と、該評価用ソフトウェアの使用に際して上記データファイルのデータを使用する場合に、該評価用ソフトウェアを使用するプロセッサから日付情報を読み取る処理と、該日付情報を読み取った後、読み取った日付情報に依存したキーに基づいて上記検証パターンの複合化を行い、上記データファイルのデータについての複合化が正しく行えるか否かを検証する処理と、上記データファイルのデータについての複合化が正しく行えないと検証された場合に、上記データファイルの使用を制限する処理

とを備えてなることを上記課題の解決手段とした。

【0011】この発明によれば、データファイルに格納するデータの少なくとも一部を、該データの作成日に依存したキーに基づき符号化して格納するとともに、この格納したデータについての複合化が正しく行えるか否かの検証を可能にした検証パターンを上記データファイルに格納しておく。そして、該評価用ソフトウェアの使用に際して上記データファイルのデータを使用する場合に、プロセッサから読み取った日付情報に依存したキーに基づいて上記検証パターンの複合化を行い、上記データファイルのデータについての複合化が正しく行えるか否かを検証する。さらに、上記データファイルのデータについての複合化が正しく行えないと検証された場合に、上記データファイルの使用を制限する処理を行う。したがって、上記データファイルの継続的な不正使用を防止することが可能になるとともに、データの少なくとも一部を符号化することにより、このデータの改ざんが行いにくくなる。

【0012】すなわち、単純に使用期限情報をデータファイルに付加しただけでは、リバースエンジニアリングなどの技術によって使用期限情報の改ざんが行われるおそれがある。このようなおそれを回避するため、例えば使用期限情報を暗号化する方法も考えられるが、その場合には、保存場所が特定されてしまうと、他の使用期限のデータファイルからその部分だけの複製を行うなどといった不正が行われるおそれがある。しかして、この発明では上述したごとくデータの改ざんを行いにくくしたことにより、評価用ソフトウェアの不正使用をより確実に防止し得るのである。

【0013】

【発明の実施の形態】以下、本発明をその実施形態例に基づいて詳しく説明する。図1(a)、(b)は本発明における請求項1記載の評価用ソフトウェアの不正使用防止方法の第1実施形態例を説明するための図であり、図1(a)は評価用ソフトウェアとなるアプリケーションファイル6の構成図、図1(b)はデータファイル7の構成図である。アプリケーションファイル6は、アプリケーションソフトウェア本体10と、プロテクトプログラム11とからなっており、アプリケーションソフトウェア本体10にはファイル管理手段12が備えられている。ファイル管理手段12は、データをファイルとして格納する手段であり、データの入出力を行うため多くのソフトウェアに備えられているものと同種の機構からなるものである。

【0014】プロテクトプログラム11には、時間獲得手段13と、データ符号化手段14と、データ複合化手段15とが備えられている。時間獲得手段13は日付(時間)の情報を獲得するためのもので、図11に示した時間取得手段5と同様にプログラムが起動したときの開始時間を取得するものであり、オペレーティングシス

テム(プロセッサ)の時間獲得手段を介してハードウェアの時計機構などから時間情報を得るものである。データ符号化手段14は、格納すべきデータをファイルの形式に変換するものであり、具体的には、格納データに数値化した日付を単純に付加するものである。データ複合化手段15は、符号化手続きが施されたデータから日付情報を取り出すものである。そして、これら時間獲得手段13、データ符号化手段14、データ複合化手段15を備えたことによってプロテクトプログラム11は、データファイル内のデータが期限内であるか否か、すなわちデータファイルが使用期限を過ぎた不正なものでないかを判定することができるものとなっている。一方、データファイル7には、図1(b)に示すように利用者データ16の他に、使用期限情報を格納する領域となる終了期限情報17が設けられている。

【0015】次に、このような構成のアプリケーションファイル6、データファイル7を用いて、データファイル7の利用者データ16にデータを格納する場合、さらにデータファイル7を読み込む場合について、図2

(a)、(b)を参照して説明する。データを格納するには、まず、図2(a)に示すようにデータファイル7が新規ファイルであるか否かをプロテクトプログラム11で判定し(ステップA1、以下S T-A1と記す)、新規ファイルであれば、時間獲得手段13にて日付を獲得するとともに、予め設定された許容使用時間(許容使用日数)を加えたものを使用期限とし、使用期限情報を作成する(S T-A2)。また、データファイル7が既存ファイルであれば、既に読み込んだ使用期限情報をそのまま用いる(S T-A3)。

【0016】次に、得られた使用期限情報をデータファイル7の終了期限情報17に付加し、これを格納ファイルとする(S T-A4)。次いで、得られた格納ファイルを、アプリケーションソフトウェア本体10のファイル管理手段12を利用して、これを2次記憶手段(図示略)などに格納する(S T-A5)。例えば、新規ファイルを格納する場合、現在の日付が95/1/1(1995年1月1日の略、以下同様)であり、許容使用時間が30日であれば、95/1/31を使用期限情報として格納する。

【0017】また、このようにして格納されたデータ(データファイル7)を利用するには、まず、図2(b)に示すようにデータファイル7を読み出し(S T-B1)、このデータファイル7から使用期限情報を取り出す(S T-B2)。なお、使用期限情報が保存されていない場合には、このデータファイル7は不正使用されているものとして、その読み込みを停止する。次に、時間獲得手段13により日付情報として現在の日付を獲得する(S T-B3)。そして、得られた日付と、先にデータファイル7から得られた使用期限情報で規定された使用期限とを比較し(S T-B4)、得られた日付が

使用期限を過ぎている場合には、上記データファイルの読み込みを停止する。

【0018】例えば、現在の日付が95/2/1であり、使用しようとしているデータファイルの使用期限情報で規定された使用期限が95/1/31であれば、その読み込みが停止され、該データファイルが使用できなくなるのである。なお、得られた日付が使用期限内である場合、つまりアプリケーションファイル6の使用が正規の期間内においてなされている場合には、データファイル7の読み込みが正規になされ、アプリケーションファイル6の評価に供される。

【0019】このような評価用ソフトウェアの不正使用防止方法にあつては、プロテクトプログラム11によって使用するデータファイル7に使用期限を記憶させ、該データファイル7の読み込み時に、その使用期限が過ぎている場合に読み込みを停止させるようにしたので、データファイル7が予め設定された一定期間しか使用することができないものとなり、したがってこのデータファイル7の使用を必要とするアプリケーションファイル6の使用も、一定期間に制限することができる。

【0020】すなわち、プログラムファイル（アプリケーションファイル）に使用期限情報を格納した従来の方法では、作成したデータファイルを保存しあるいはこれの複製を保存しておくことにより、使用期限が過ぎてプログラムファイル（アプリケーションファイル）のみならずデータファイルまでが消去されたとしても、新たな評価として評価プログラムを再設定し、複製したデータファイルを開くことにより、結果的にプログラムファイル（アプリケーションファイル）の継続使用が可能になってしまう。しかして、上記本実施形態例における方法によれば、データファイルの複製を行っても、データファイル側に使用期限情報が格納されているため、使用期限が過ぎた後での該データファイルの使用を確実に制限することができるのである。

【0021】一般にソフトウェアには、諸設定のためにファイルを利用することがあり、ソフトウェアの起動には、その設定ファイルが必須である。したがって、この設定ファイルに本発明の方法を適用すれば、ソフトウェアそのものが入手してから許容された一定期間のみ動作可能とすることができる。また、本発明を複数のソフトウェアに適用することにより、複数ソフトウェア間でのデータの受け渡しの評価も行えるようにすることができる。

【0022】図3（a）、（b）は本発明における請求項1記載の評価用ソフトウェアの不正使用防止方法の第2実施形態例を説明するための図であり、図3（a）は評価用ソフトウェアとなるアプリケーションファイル18の構成図、図3（b）はデータファイル19の構成図である。この第2実施形態例が上記第1実施形態例と異なるところは、データファイルに使用期限情報を付加し

てこれに格納するに際して、該使用期限情報を、日付を単に数値化するのではなく、暗号化する点にある。暗号化については、日付情報に対する復元可能な演算であればどのような形態のものでもよく、もちろん既存の暗号化技術を適用することができる。

【0023】図3（a）に示した評価用ソフトウェアとなるアプリケーションファイル18は、図1（a）に示したアプリケーションファイル6と同様にアプリケーションソフトウェア本体20とプロテクトプログラム21とからなっており、アプリケーションソフトウェア本体20には上記ファイル管理手段12と同様のファイル管理手段22が備えられている。

【0024】プロテクトプログラム21には、時間獲得手段23と、データ符号化手段24と、データ複合化手段25と、使用期限情報26とが備えられている。時間獲得手段23は、図1（a）に示した時間獲得手段13と同様に機能するものである。また、データ符号化手段24は、上述したごとく使用期限情報を暗号化し、データファイル19に付加するものである。データ複合化手段25は、データファイル19から使用期限情報を取り出し、暗号化の逆演算を施して元の使用期限となる日付を表す情報に戻すためのものである。使用期限情報26は、最初の起動時に設定される使用期限情報を格納するためのものである。なお、この使用期限情報26には、アプリケーションファイル18自体が配布された際、未設定を示す所定の値が設定されるようになっている。また、このアプリケーションファイル18には、そのメインメモリ上のプロテクトプログラム21領域中に、該ファイル18が起動している際該ソフトウェアの実行によって開かれたファイルに対する符号化キーKiを保持する領域（図示略）が設けられている。一方、データファイル19には、図3（b）に示すように利用者データ27の他に、使用期限情報を格納する領域となる使用期限情報28が設けられている。

【0025】次に、このような構成のアプリケーションファイル18、データファイル19を用いて、データファイル19の利用者データ27にデータを格納する場合、データファイル19を読み込む場合、さらにアプリケーションファイル18の起動を確認する場合について、図4（a）、（b）、（c）を参照して説明する。データを格納するには、まず、図4（a）に示すようにデータファイル19が新規ファイルであるか否かをプロテクトプログラム21で判定し（ST-C1）、新規ファイルであれば、時間獲得手段23にて日付を獲得するとともに予め設定された許容使用時間（許容使用日数）を加えて使用期限を計算し、この使用期限と、それまでに開かれたファイルの使用期限（使用期限情報）のうち最も早期のものを正規の使用期限とし、この使用期限に基づく情報をデータ符号化手段24で暗号化する（ST-C2）。また、データファイル19が既存ファイルで

あれば、既に読み込んだ使用期限情報をそのまま用いる（ST-C3）。

【0026】次に、得られた使用期限情報をデータファイル19の使用期限情報28に付加し、これを格納ファイルとする（ST-C4）。次いで、得られた格納ファイルを、アプリケーションソフトウェア本体20のファイル管理手段22を利用して、これを2次記憶手段（図示略）などに格納する（ST-C5）。

【0027】また、このようにして格納されたデータ（データファイル19）を利用するには、図2（b）に示した場合と同様に、まず、図4（b）に示すようにデータファイル19を読み出し（ST-D1）、このデータファイル19から使用期限情報を取り出す（ST-D2）。なお、使用期限情報が保存されていない場合には、このデータファイル19は不正使用されているものとして、その読み込みを停止する。次に、時間獲得手段23により日付情報として現在の日付を獲得する（ST-D3）。そして、得られた日付と、先にデータファイル19から得られた使用期限情報で規定された使用期限、すなわちデータ複合化手段25で暗号化の逆演算が施されて得られた元の使用期限とを比較し（ST-D4）、得られた日付が使用期限を過ぎている場合には、上記データファイルの読み込みを停止してその使用を制限し、すなわちこの例では、データファイル19とアプリケーションファイル18とを消去してこれらの再使用を不能にする（ST-D5）。なお、得られた日付が使用期限内である場合、つまりアプリケーションファイル18の使用が正規の期間内においてなされている場合には、図2（b）に示した場合と同様にデータファイル19の読み込みが正規になされ、アプリケーションファイル18の評価に供される。

【0028】また、アプリケーションファイル18を起動させる場合には、その起動の確認として、まず、図4（c）に示すようにアプリケーションファイル18のプロテクトプログラム21から使用期限情報26を取り出す（ST-E1）。このとき、この使用期限情報26が未設定であるか否かを判断し（ST-E2）、未設定である場合には後述するST-E5に処理を進める。使用期限情報26が設定されている場合には、時間獲得手段23により日付情報として現在の日付を獲得する（ST-E3）。そして、得られた日付と、上記使用期限情報26に設定された使用期限とを比較し（ST-E4）、得られた日付が使用期限を過ぎている場合には、上記データファイルの読み込みを停止してその使用を制限し、すなわちこの例では、データファイル19とアプリケーションファイル18とを消去してこれらの再使用を不能にする（ST-E6）。

【0029】なお、得られた日付が使用期限内である場合、つまりアプリケーションファイル18の使用が正規の期間内においてなされている場合には、アプリケーシ

ョンファイル18の起動が正規になされる。また、使用期限情報26が未設定である場合には、時間獲得手段23にて日付を獲得するとともに予め設定された許容使用時間（許容使用日数）を加えて使用期限を計算し、得られた使用期限を使用期限情報26に格納し（ST-E5）た後、アプリケーションファイル18を起動させる。

【0030】このような評価用ソフトウェアの不正使用防止方法にあつては、データファイル19の使用期限情報28により、一定期間を過ぎたデータ（利用者データ27）およびプログラム（アプリケーションファイル18）が動作しないことから、一定期間が過ぎた評価用ソフトウェアの使用を防止することができる。このとき、アプリケーションファイル18を自動的に消去することにより、より確実に不正使用を防止することができる。また、データファイルに単純に使用期限情報を付加しただけでは、さまざまなツールによってその使用期限情報の改ざんがなされるおそれがあるものの、上記実施形態例では、データファイル19に格納する使用期限情報に暗号化を施しているため、リバースエンジニアリングなどデータファイルの解析によって使用期限情報の改ざんがなされるのを防止することができる。

【0031】さらに、複数のファイルを同時に開くことのできるソフトウェアの場合、新規データファイルの使用期限として、それまでに開かれたデータファイルの各々の使用期限の最も早期のものを採用するため、ソフトウェア内での複製による実質的なデータの使用期間延長の試みを防止することができる。このとき、データファイルに該ソフトウェアの起動に必須な設定ファイルを含めれば、これが常に読まれることから特に有効となる。なお、本実施形態例では、新規データファイルの場合のみ既存データファイルの使用期限情報との間で比較・置き換えを行うようにしたが、既存データファイルについても、他の既存データファイルの使用期限情報との間で比較・置き換えを行うようにしてもよい。

【0032】図5（a）、（b）は本発明における請求項5記載の評価用ソフトウェアの不正使用防止方法の第1実施形態例を説明するための図であり、図5（a）は評価用ソフトウェアとなるアプリケーションファイル30の構成図、図5（b）はデータファイル31の構成図である。この実施形態例が図1、図3に示した実施形態例と異なるところは、使用期間に相当する日付毎に、キーを変えてデータファイルを符号化する点である。ここで、符号化はデータファイルの全体に対してキーをパラメータとした可逆な演算とし、演算結果からキーが推論困難であればどのようなものでもよく、既存の暗号化技術や圧縮技術を適用してもよい。

【0033】図5（a）に示したアプリケーションファイル30は、アプリケーションソフトウェア本体32と、プロテクトプログラム33とからなっており、アプ

リケーションソフトウェア本体32にはファイル管理手段34が備えられている。プロテクトプログラム33には、時間獲得手段35と、キー作成手段36と、データ符号化手段37と、データ複合化手段38とが備えられている。時間獲得手段35は、図1(a)に示した時間獲得手段13と同様に日付(時間)の情報を獲得するためのものである。キー作成手段36は、時間獲得手段35で得られた日付情報を基に、符号化のためのキーを計算するものである。ここで、符号化のためのキーとしては、後述するように一定期間の日付に対しては同じ値であり、期間外では異なるように形成する。データ符号化手段37は、格納すべきデータファイルを符号化してデータファイル31に格納するとともに、この格納したデータについての複合化が正しく行えるか否かの検証を可能にした検証パターンを該データファイル31に格納するものである。データ複合化手段38は、データ符号化手段37によって符号化手続きが施されたデータを複合化するためのものであり、すなわち、符号化されたデータを逆演算することによって元の形態に復元するものである。一方、データファイル31には、図5(b)に示すように符号化された利用者データ39が格納されるようになっている。

【0034】そして、このような構成によりプロテクトプログラム33は、データ符号化手段37によってデータを符号化するとともに格納した検証パターンが、データ複合化手段38によって正しく復元されているか否かを確認することにより、データが使用期限内のものであるか否か、すなわち不正使用のファイルであるか否かを判定し得るものとなっている。ここで、上記符号化のキーについては、図6に示すようにデータの作成日に依存して一定期間毎(図中の例では30日毎)に変更される。このようにして符号化のキーを変更すると、同じデータファイルでも、作成日が異なる期間である場合符号化のされ方も変わってくる。したがって、符号化された日(作成日)と同一の期間内だけ、該データファイルが正しく復元できるようになるのである。

【0035】次に、このような構成のアプリケーションファイル30、データファイル31を用いて、データファイル31の利用者データ39にデータを格納する場合、さらにデータファイル31を読み込む場合について、図7(a)、(b)を参照して説明する。データを格納するには、まず、図7(a)に示すようにデータファイル31が新規ファイルであるか否かをプロテクトプログラム33で判定し(ST-F1)、新規ファイルであれば、時間獲得手段13にて日付を獲得する(ST-F2)とともに、獲得した日付から符号化キーを作成する(ST-F3)。一方、新規ファイルでなく既存ファイルである場合には、既に読み込まれた符号化キーをそのまま利用する(ST-F4)。

【0036】次に、格納すべきデータに検証パターンを

付加し、さらに上記符号化キーを用いた符号化を行う(ST-F5)。次いで、符号化されたデータファイル31を、アプリケーションファイル30のファイル管理手段34を利用して、これを2次記憶(図示略)などに格納する(ST-F6)。

【0037】ここで、日付から符号化のキーを作成する処理については、前述したように作成されたものが一定期間同じであり、以後のものとは異なるという性質を持つてばどのようなものでもよいが、日付を、例えば1日が1に対応するような数値で表現し、それを期間(例えば日数)で整数除算を行い、その値をハッシュ化するなどといった方法が考えられる。例えば、新規ファイルを格納する場合、現在の日付が95/1/10であり、許容使用期限が30日であるとする。そして、95/1/10を数値化したものが仮に32882であれば、これを30で割って整数部をとると、1108となる。この値にもう一度ハッシュ関数などによる変換を施し、 $h(1108) = 104325$ なるキーを得る。なお、ハッシュ関数を施す理由は、キーが近い数値であると、符号化の結果も変化が乏しくなりパターンによる判定が困難になる可能性があるからである。

【0038】また、このようにしてデータファイル31に保存されたデータを利用するためには、まず、図7(b)に示すようにデータファイル31を読みだす(ST-G1)。次に、時間獲得手段35によって現在の日付を獲得し(ST-G2)、獲得した日付に基づき符号化のときと同様にしてキーを定める(ST-G3)。次いで、定めたキーを用いてデータ複合化手段38による複合化を行い、先に付加した検証パターンを得る(ST-G4)。その後、複合化されたものが、定められた検証パターンを保持しているか否か、すなわち得られた検証パターンが正規のものであるか否かを判断し(ST-G5)、正規のものであれば読み込み処理を継続し、正規のものでなければ読み込み処理を停止する。

【0039】例えば、現在の日付が95/2/1であるとする、その数値化したものが33269となり、これを30で割って得られた値の整数部をとると1108となる。この値にハッシュ関数などによる変換を施すと、 $h(1108) = 104325$ が得られる。この値は、上記したように95/1/10に符号化したのと同じキーとなり、したがってデータの複合化が正常に行え、これにより検証パターンも正常に復元されるので、後の処理を正常に行うことができる。

【0040】一方、現在の日付が95/2/2であるとする、その数値化したものが33270となり、これを30で割って整数部をとると1109となる。この値にハッシュ関数などによる変換を施すと、得られるキーは例えば $h(1109) = 600455$ となる。したがって、これをキーとしてデータや検証パターンの複合化を試みると、当然ながら正しく複合化することができ

ず、もちろん検証パターンについてもこれを正しく復元することができない。したがって、不正な期間に用いられているデータであると判定されるのである。

【0041】このような評価用ソフトウェアの不正使用防止方法にあつては、プロテクトプログラム33によって符号化のキーを日付（作成日）により変化させているので、キーの一致する期間でなければ符号化されたデータファイルを利用できないようにすることができる。すなわち、プログラムファイル（アプリケーションファイル）に使用期間情報をもつ従来の方法では、前述したように作成したデータファイルを保存しあるいはこれの複製を保存しておくことにより、結果的にプログラムファイル（アプリケーションファイル）の継続使用が可能になってしまうものの、本方法によれば、データファイル31の複製を行っていても、使用期限を過ぎている場合にはこれを起動した日付情報に基づいて複合化しても正規な複合化が行えないことから、得られるデータが正規のものでなく解読できないものとなるので、結果として該データファイル31、すなわち評価用ソフトウェアそのものの使用期間外での使用を制限することができる。

【0042】また、この方法によれば、評価ソフトウェアの設定（インストール）を利用者に委ねても不正使用されるおそれがないため、評価用のソフトウェアをネットワーク上で一般に公開したり、CD-ROMなどで配布するなどすることにより、メーカー側の設定などに伴う保守作業等を軽減することができる。また、前述したようにソフトウェアには、諸設定のためにファイルを利用することがあり、ソフトウェアの起動には、その設定ファイルが必須である。したがって、この設定ファイルに本方法を適用すれば、ソフトウェアそのものを、入手してから許容された一定期間のみ動作可能とすることができる。さらに、本方法を複数のソフトウェアに適用することにより、複数ソフトウェア間でのデータの受け渡しの評価も行うようにすることができる。また、本方法によれば、データを符号化するため、データ解析ツールなどによって解析を行おうとしてもその内容が解析しにくくなり、したがって評価用ソフトウェアの不正使用をより確実に防止することができる。

【0043】図8（a）、（b）は本発明における請求項5記載の評価用ソフトウェアの不正使用防止方法の第2実施形態例を説明するための図であり、図8（a）は評価用ソフトウェアとなるアプリケーションファイル40の構成図、図8（b）はデータファイル41の構成図である。この実施形態例が図5に示した実施形態例と異なるところは、図5に示した実施形態例によると、評価用ソフトウェアで作成されたデータが一定期間のみ使用できるようになるものの、例えば2/2で一つの使用期限がくる場合に、2/1に評価を開始すると、そこで作成したデータが1日しか使用できなくなるといったように、キーの変更になる期日近くに評価を開始した場合

に、作成したデータがごく短い期間でしか利用できなくなるといったことを改善した点にある。

【0044】図8（a）に示したアプリケーションファイル40は、図5（a）に示したアプリケーションファイル30と略同様にアプリケーションソフトウェア本体42とプロテクトプログラム43とからなるものである。一方、図8（b）に示したデータファイル41は、符号化された利用者データ44が格納されるようになっているとともに、期間の種別を格納する領域となる期間の種別45が設けられている。ここで、期間の種別45に関しては、符号化が施されていないものとする。期間の種別45は、本実施形態例においては分割された期間を交互に、AもしくはBの種別を割り当てるものとする。この種別は、例えば日付の数値データを単位期間で整数除算した値が偶数であるか奇数であるかなどの方法により、日付情報から定めることができる。

【0045】また、このアプリケーションファイル40には、そのメインメモリ上のプロテクトプログラム43領域中に、該ソフトウェア（アプリケーションファイル40）が起動している際該ソフトウェアの実行によって開かれたファイルに対する符号化キー*K_i*を保持する領域（図示略）が設けられている。さらに、プロテクトプログラム43中には、現在のキーおよびその一つ前の期間のキーと、それぞれの期間種別とを格納する領域（図示略）が設けられている。そして、このような構成に基づき本実施形態例では、データの複合化の際、該当する日付に基づくキーの他に、一つ前の期間のキーを用いた複合化をも試みることにより、評価の開始時点から二つの期間に亘って使用できるようにしている。このようにすれば、最短でも予め設定された一つの期間内でフルに評価を行うことができ、また最長では二つの期間内でフルに評価を行うことができるようになる。なお、データ複合化の試みを効率的にするため、本実施形態例では期間を大きく2種類に分け、どちらのキーを利用するかを識別するためのタグを設けている。

【0046】すなわち、先の第1実施形態例にあつては、図9（a）に示すように期間T11で読み込み動作を行うプログラムP11では該期間T11内に作成されたデータに対応した複合化しか行えないため、該期間T11内に作成されデータ（例えばD11）しか利用できなくなる。しかして、本実施形態例では、図9（b）に示すように一つ前の符号化のキーも計算してこのキーに基づくデータの複合化をも行えるようにすることから、一つ前の期間に作成したデータも利用可能となるのである。さらに、複数のファイルを同時に開くことのできるソフトウェア（アプリケーションファイル40）の場合には、開かれたデータファイル41に設定されているキーのうちの最も短い使用期限をもつものを新規ファイルのためのキーとして符号化することにより、ソフトウェア内での複製による使用期限の延長を防ぐこともできる

ようになっている。

【0047】次に、このような構成のアプリケーションファイル40、データファイル41を用いて、データファイル41の利用者データ44にデータを格納する場合、さらにデータファイル41を読み込む場合について、図10(a)、(b)を参照して説明する。データを格納するには、まず、図10(a)に示すようにデータファイル31が新規ファイルであるか否かをプロテクトプログラム33で判定し(ST-H1)、新規ファイルであれば、時間獲得手段35にて日付を獲得するとともに、獲得した日付から期間種別を計算し(ST-H2)、さらに獲得した日付から符号化キーを作成する(ST-H3)。ただし、それ以前に、獲得した日付が含まれる期間より一つ前の期間にデータファイルを開いていれば、一つ前の期間の符号化キーおよび期間種別を用いる。また、既存ファイルであれば、読み込んだ符号化キーおよび期間種別を利用する(ST-H4)。

【0048】次に、格納すべきデータに検証パターンを付加し、さらに上記符号化キーを用いた符号化を行い、さらに得られた期間種別の情報を付加する(ST-H5)。次いで、符号化されたデータファイル41を、アプリケーションファイル40のファイル管理手段34を利用して、これを2次記憶(図示略)などに格納する(ST-H6)。例えば、新規ファイルを格納する場合、現在の日付が95/1/10であり、許容使用期限が最大60日であり、期間の最小単位が30日とする。そして、95/1/10を数値化したものが、仮に32882であれば、これを30で割って整数部をとると、1108となる。この値にもう一度ハッシュ関数などによる変換を施し、 $h(1108) = 104325$ なるキーを得る。ここで、1108は偶数なので、期間種別はAとする。

【0049】また、このようにしてデータファイル41に保存されたデータを利用するためには、まず、図10(b)に示すようにデータファイル41を読み出す(ST-I1)。次に、時間獲得手段35によって現在の日付を獲得し、さらにその期間種別を定める(ST-I2)。そして、定めた期間種別と先にデータファイル41に格納した期間種別とを比較し(ST-I3)、等しい場合には獲得した現在の日付を用いて符号化キーを定める(ST-I4)。また、異なる場合には、一つ前の期間のキー、期間種別をそれぞれ計算し、複合化のキーとする(ST-I5)。次いで、定めたキーを用いてデータ複合化手段38による複合化を行い、先に付加した検証パターンを得る(ST-I6)。その後、複合化されたものが、定められた検証パターンを保持しているか否か、すなわち得られた検証パターンが正規のものであるか否かを判断し(ST-I7)、正規のものであれば読み込み処理を継続し、正規のものでなければ読み込み処理を停止する。

【0050】例えば、現在の日付が95/2/2であるとする、その数値化したものが33269となり、これを30で割って整数部をとると1109となる。この値は奇数であるので、期間種別をBと認識する。もし、期間種別Aのファイルを開こうとした場合、期間種別が異なるため1期間前のキーである、 $h(1109-1) = 104325$ が用いられる。すると、95/1/10に符号化したデータを用いることができることから正常に複合化を行うことができ、したがって検証パターンも正常に復元されるので、処理を正常に行うことができる。ここで、この例では許容日数が60日になっている。また、95/3/5は数値化したものが33301となり、30で割って整数部をとると1110となり、期間種別がAとなる。期間種別Aのファイルでキーは、例えば $h(1110) = 775254$ となる。したがって、これをキーとして1/1作成のファイル(期間種別はA)の複合化を試みると、当然ながら正しく複合化できず、検証パターンも正しく復元しない。よって、不正な期間に用いられているデータであると判定されるのである。

【0051】このような評価用ソフトウェアの不正使用防止方法にあつては、先の図5に示した実施形態例と同様の効果が得られる。さらに、評価開始時点で作成されたデータが、ひとつ先の単位期間まで有効であるため、少なくとも最小の単位期間についてはその使用が可能となり、したがって使用期間が極端に短くなり、評価を行う時間が十分に確保されなくなる不都合を回避することができる。また、新規データファイルの符号化キーとして、それまでに開かれたデータファイルの各々の使用期限のうち最も早期のものを採用するため、ソフトウェア(アプリケーションファイル40)内でのファイル間のデータの複製による実質的なデータの使用期間延長の試みを防止することができる。このとき、データファイルに該ソフトウェアの起動に必須な設定ファイルを含めれば、これが常に読み込まれることから特に有効となる。

【0052】なお、本実施形態例では、一つ前の単位期間のデータまで有効としたが、二つ前までや三つ前までなど、任意の期間数だけ遡って許容するように設定してもよい。また、符号化を期間種別以外のデータファイルに対して行っているが、期間種別を含まない任意のデータファイルの一部に限定して行うようにしてもよい。さらに、プロテクトプログラム43については、図8

(a)に示したようにデータ符号化手段37を含む構成としたが、書き込みの不要なソフトウェアに対しては、該データ符号化手段37をもたないように構成することも可能であり、その場合には、データは予めその一部もしくは全部が符号化されて提供されるようにすればよい。

【0053】

【発明の効果】以上説明したように本発明における請求

項 1 記載の評価用ソフトウェアの不正使用防止方法は、利用者の作成するデータファイルに使用期限情報を付加し、さらにこの使用期限情報とプロセッサから読み取った日付情報とを比較し、使用期限を過ぎている場合に上記データファイルの使用を制限する処理を行うようにした方法であるから、上記データファイルの継続的な不正使用を防止することができ、これにより本質的にデータの継続使用が必須の評価用ソフトウェアの継続的な不正使用を防止することができる。

【0054】本発明における請求項 5 記載の評価用ソフトウェアの不正使用防止方法は、データファイルに格納するデータの少なくとも一部を、該データの作成日に依存したキーに基づき符号化して格納するとともに、この格納したデータについての複合化が正しく行えるか否かの検証を可能にした検証パターンを上記データファイルに格納し、該評価用ソフトウェアの使用に際して上記データファイルのデータを使用する場合に、プロセッサから読み取った日付情報に依存したキーに基づいて上記検証パターンの複合化を行い、上記データファイルのデータについての複合化が正しく行えるか否かを検証するようにし、上記データファイルのデータについての複合化が正しく行えないと検証された場合に、上記データファイルの使用を制限するようにした方法であるから、上記データファイルの継続的な不正使用を防止できるとともに、データの少なくとも一部を符号化することにより、このデータの改ざんが行いにくくすることができる。

【0055】このように本発明にあっては、データファイルの使用期間を制限するようにしたことから、特にデータベースなど、データ自体が重要であるソフトウェアの応用分野では、その本格的な使用に同じデータを長期に亘って利用することが必須となることから極めて有効となる。また、アプリケーションファイル（ソフトウェア）自体に設定データなど必須のデータを書き込む場合、該データの使用を制限（不能）することにより、ソフトウェアの不正使用を防止することができる。さらに、パッケージプログラムなどソフトウェアプログラムだけでなく、電子本や M I D I データなどで、一定の読み取りソフトウェアを想定し、データを販売するものに対しても、その宣伝、評価用として一般に配布する場合に本発明を適用することができ、その場合に、販売元が購入前の利用者に一定期間、音楽や情報を試用させることが可能になる。また、評価に限らず、一般にデータの試用期間を限定したい用途に応用することができ、例えば、学習教材のソフトウェアや雑誌の懸賞クイズにおいて、その正解部分を本発明により一定の時期になるまで参照できないようにするようなこともできる。

【図面の簡単な説明】

【図 1】請求項 1 記載の発明における第 1 実施形態例を説明するための図であり、（a）はアプリケーションフ

ァイルの構成図、（b）はデータファイルの構成図である。

【図 2】請求項 1 記載の発明における第 1 実施形態例を説明するためのフロー図であり、（a）はデータファイルを格納する場合についてのフロー図、（b）はデータファイルを読み込む場合についてのフロー図である。

【図 3】請求項 1 記載の発明における第 2 実施形態例を説明するための図であり、（a）はアプリケーションファイルの構成図、（b）はデータファイルの構成図である。

【図 4】請求項 1 記載の発明における第 2 実施形態例を説明するためのフロー図であり、（a）はデータファイルを格納する場合についてのフロー図、（b）はデータファイルを読み込む場合についてのフロー図、（c）はアプリケーションファイルの起動を確認する場合についてのフロー図である。

【図 5】請求項 5 記載の発明における第 1 実施形態例を説明するための図であり、（a）はアプリケーションファイルの構成図、（b）はデータファイルの構成図である。

【図 6】符号化のキーについての説明図である。

【図 7】請求項 5 記載の発明における第 1 実施形態例を説明するためのフロー図であり、（a）はデータファイルを格納する場合についてのフロー図、（b）はデータファイルを読み込む場合についてのフロー図である。

【図 8】請求項 5 記載の発明における第 2 実施形態例を説明するための図であり、（a）はアプリケーションファイルの構成図、（b）はデータファイルの構成図である。

【図 9】データファイルと実行中のプログラムとの関係を示す図であり、（a）は請求項 5 記載の発明における第 1 実施形態例の場合の関係説明図、（b）は請求項 5 記載の発明における第 1 実施形態例の場合の関係説明図である。

【図 10】請求項 5 記載の発明における第 2 実施形態例を説明するためのフロー図であり、（a）はデータファイルを格納する場合についてのフロー図、（b）はデータファイルを読み込む場合についてのフロー図である。

【図 11】従来のアプリケーションファイルの構成図である。

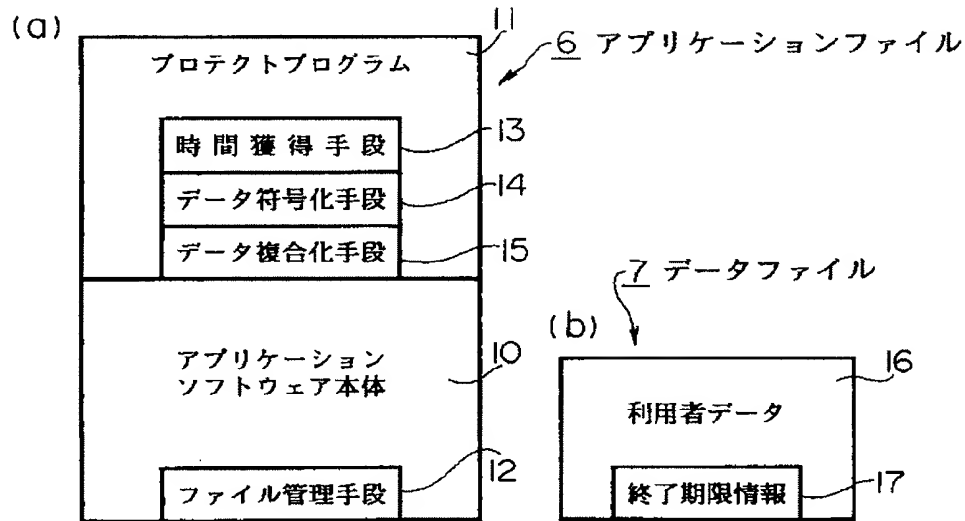
【符号の説明】

- 6、18、30、40 アプリケーションファイル
- 7、19、31、41 データファイル
- 10、20、32、42 アプリケーションソフトウェア本体
- 11、21、33、43 プロテクトプログラム
- 12、22、34 ファイル管理手段
- 13、23、35 時間獲得手段
- 14、24、37 データ符号化手段
- 15、25、38 データ複合化手段

16、27、39、44 利用者データ
17 終了期限情報

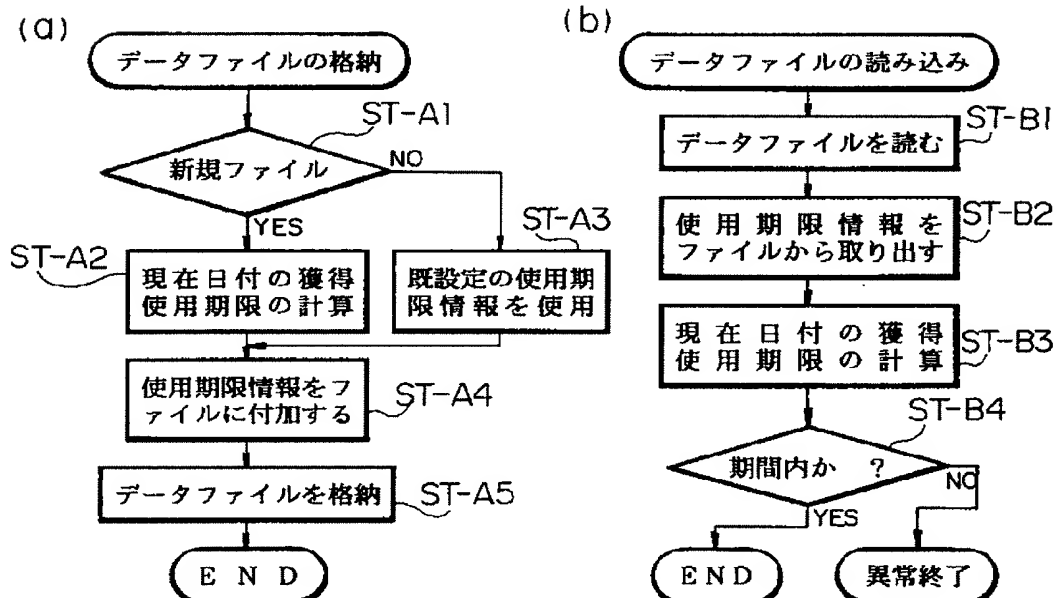
26、28 使用期限情報
45 期間の種別

【図1】



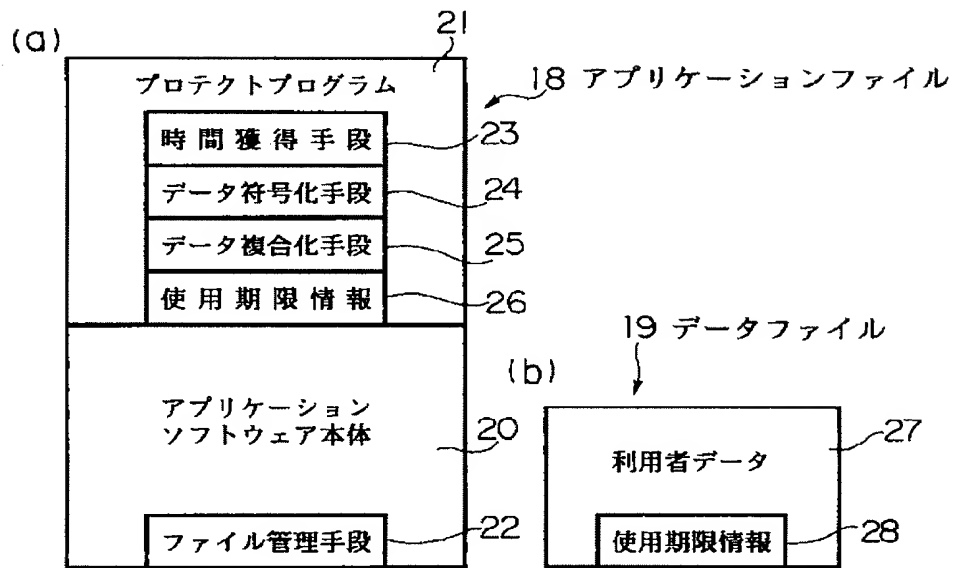
実施形態例の説明図

【図2】



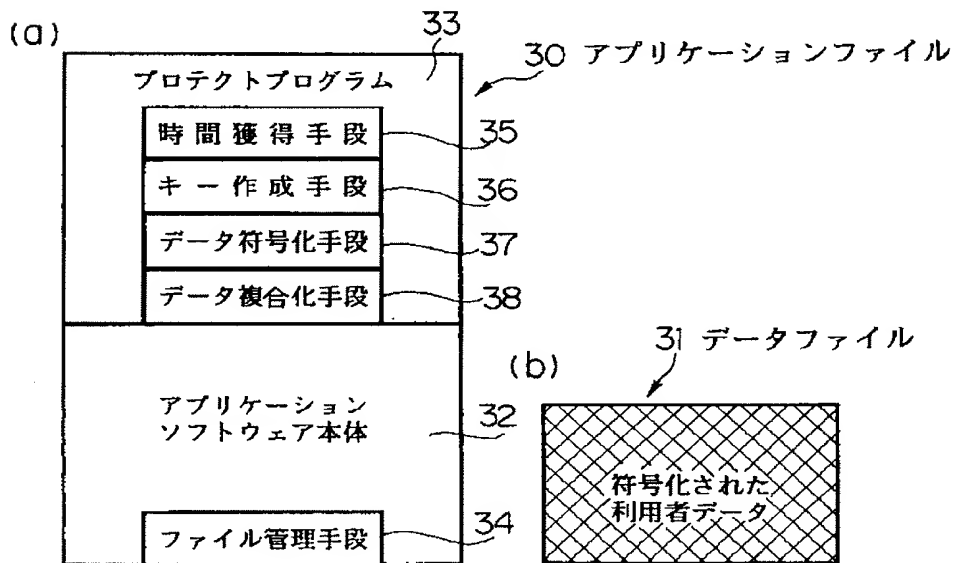
実施形態例のフロー図

【図3】



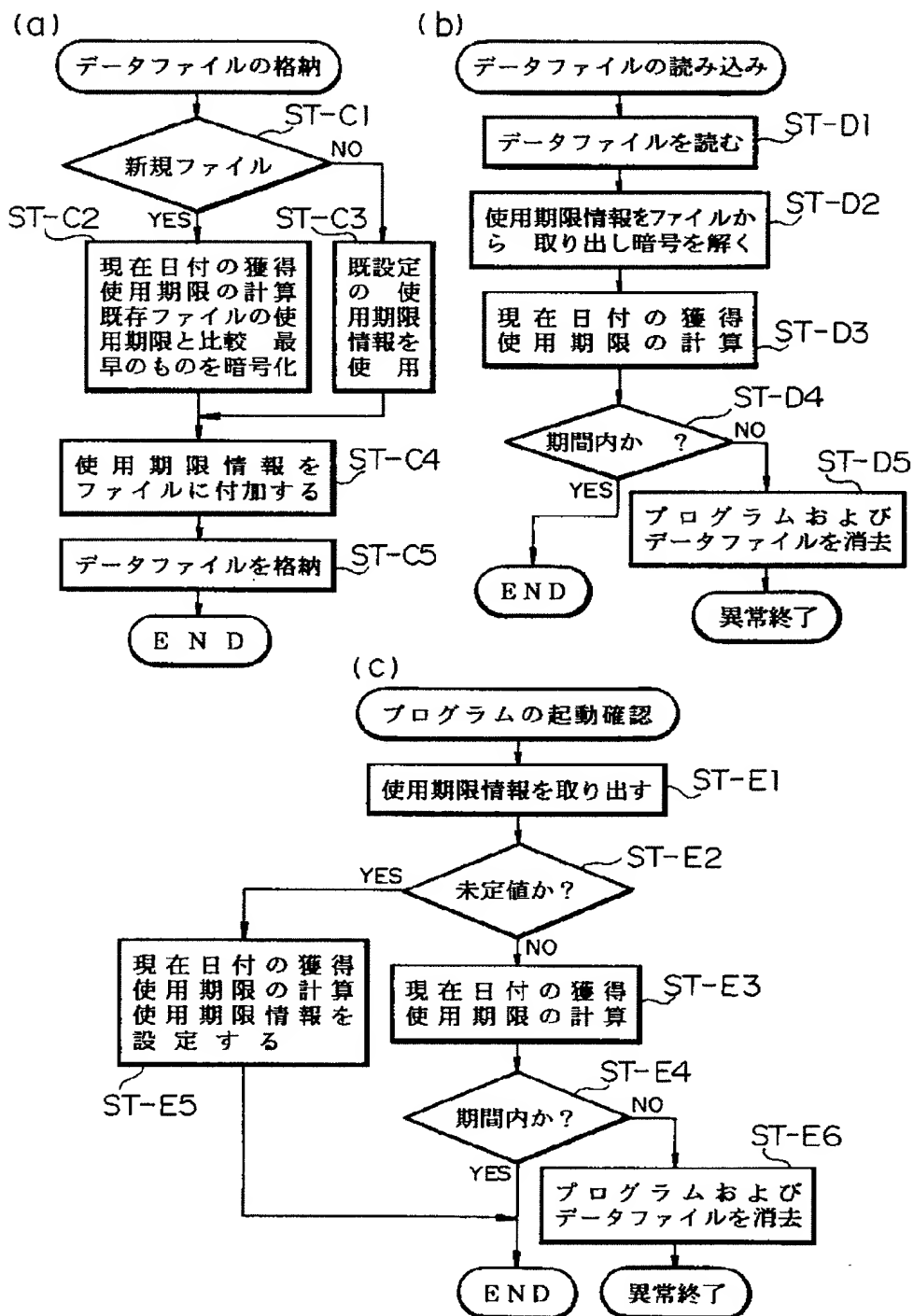
実施形態例の説明図

【図5】






実施形態例の説明図

【図4】



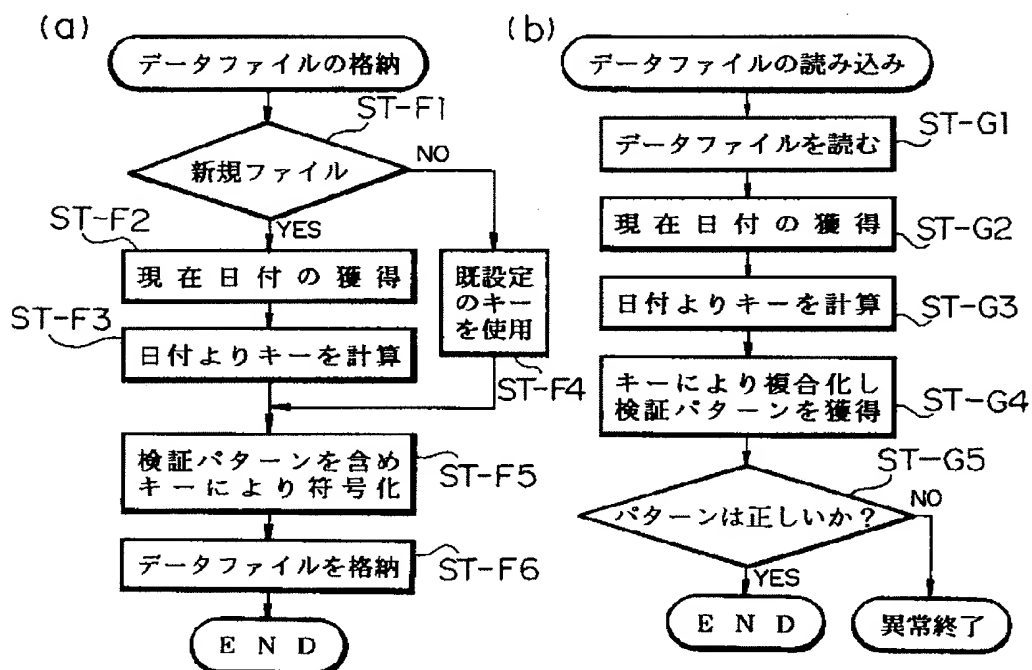
実施形態例のフロー図

【図6】

日付	95/1/1	2/1	3/1	4/1
数値化	33238			
日付/30	1107	1108	1109	1110
キー	104325	600455	775254	84652
符号化された データファイル				

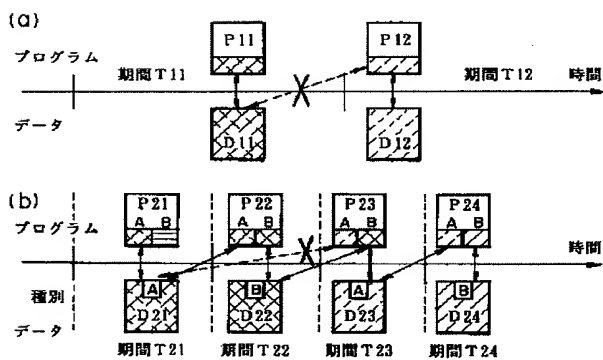
符号化キーの説明図

【図7】



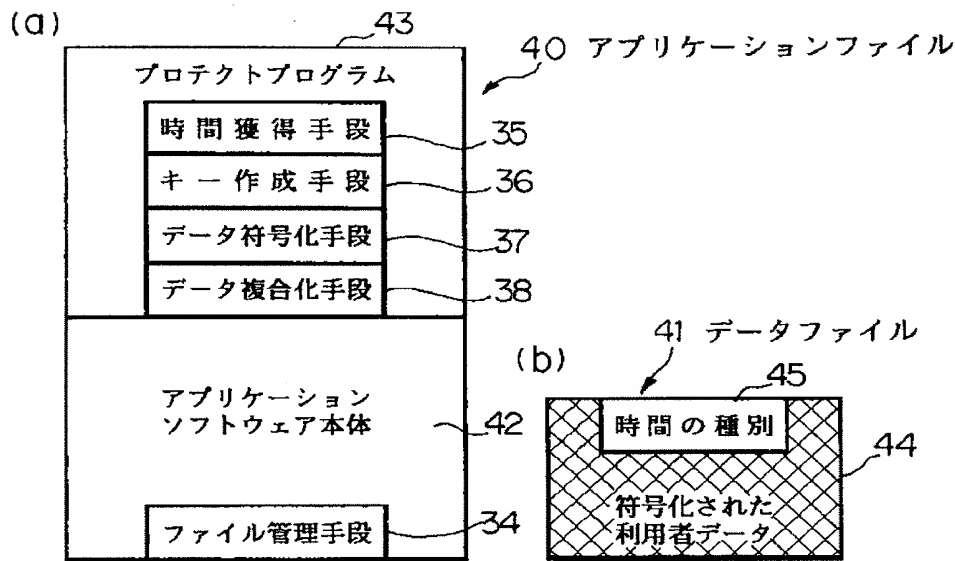
実施形態例のフロー図

【図9】



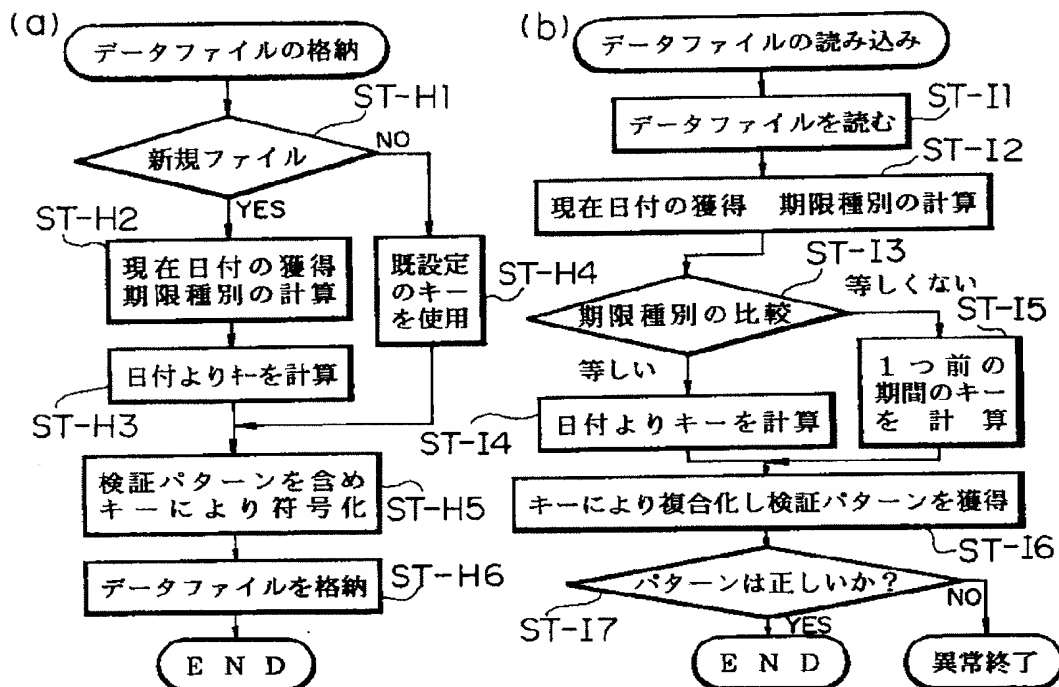
データとプログラムとの関係説明図

【図8】



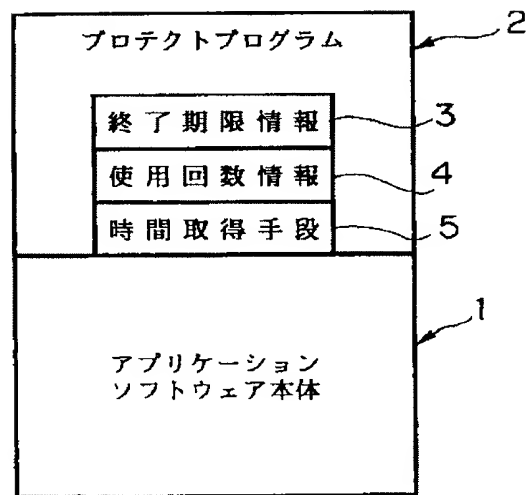
実施形態例の説明図

【図10】



実施形態例のフロー図

【図11】



従来例の説明図

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-179734

(43)Date of publication of application : 11.07.1997

(51)Int.Cl.

G06F 9/06

G06F 12/14

(21)Application number : 07-338392 (71)Applicant : OKI ELECTRIC IND CO LTD

(22)Date of filing : 26.12.1995 (72)Inventor : MORITA KOHAKU

(54) METHOD FOR PREVENTING UNAUTHORIZED USE OF SOFTWARE FOR EVALUATION

(57)Abstract:

PROBLEM TO BE SOLVED: To prevent the continuous unauthorized use of a data file by comparing use time limit information and date information read from a processor and limiting the use of the data file in the case of exceeding a use time limit.

SOLUTION: A date is acquired in a time acquisition means 13 the one to which allowable using time set beforehand is added is defined as the use time limit and the use time limit information is prepared. Then the obtained use time limit information is added to the end time limit information 17 of the data file 7 and it is turned to a storage file. Then the obtained storage file is stored in a secondary storage means or the like by utilizing the file management means 12 of an application software main body 10. Then the date at present is acquired as the date information by the time acquisition means 13. Then the obtained date is compared with the use time limit stipulated by the use time limit information previously obtained from the data file 7 and the read of the data file is stopped in the case that the obtained date exceeds the use time limit.

CLAIMS

[Claim(s)]

[Claim 1] Processing which is an unauthorized use prevention method of software for evaluation which needs use of a data file adds expiration date information to the above-mentioned data file and is stored in this Processing which reads a day entry in a processor which uses this software for evaluation Expiration date information stored in the above-mentioned data file is compared with a day entry read in the above-mentioned processor An unauthorized use prevention method of software for evaluation which is provided with processing which restricts use of the above-

mentioned data file when a date obtained from this day entry has passed over the expiration date specified using expiration date information and is characterized by things.

[Claim 2] An unauthorized use prevention method of the software for evaluation according to claim 1 adding expiration date information to the above-mentioned data file facing storing in this and enciphering this expiration date information.

[Claim 3] It faces that the above-mentioned data file adds expiration date information to a new data file which has not added those with two or more and expiration date information. An unauthorized use prevention method of the software for evaluation according to claim 1 adding and storing expiration date information so that the expiration date specified using this expiration date information may turn into the same expiration date as the earliest thing when there is a data file in which expiration date information is already added.

[Claim 4] An unauthorized use prevention method of the software for evaluation according to claim 1 wherein processing which restricts use of the above-mentioned data file is the processing which makes use of this software for evaluation impossible by eliminating a program file and a data file of software for evaluation.

[Claim 5] It is an unauthorized use prevention method of software for evaluation which needs use of a data file. While coding based on a key which depended on the Date of drafting of this data for at least some data stored in the above-mentioned data file and storing this coded data in the above-mentioned data file after that. Processing which stores a verification pattern which enabled verification of whether to be able to perform composite-ization about this stored data correctly in the above-mentioned data file. Processing which reads a day entry in a processor which uses this software for evaluation when using data of the above-mentioned data file when using this software for evaluation. Processing which verifies whether composite-ization of the above-mentioned verification pattern is performed based on a key depending on a read day entry and composite-ization about data of the above-mentioned data file can be correctly performed after reading this day entry. An unauthorized use prevention method of software for evaluation which is provided with processing which restricts use of the above-mentioned data file when composite-ization about data of the above-mentioned data file could not be performed correctly and it is verified and is characterized by things.

[Claim 6] Processing which verifies whether composite-ization about data of the above-mentioned data file can be performed correctly. When using data of the above-mentioned data file when using this software for evaluation. In addition to a key depending on a day entry read in the above-mentioned processor 1 time. Or an unauthorized use prevention method of the software for evaluation according to claim 5 being the processing which prepares a key depending on a day entry which traced back a period of the defined number of times and performs composite-ization of the above-mentioned verification pattern based on these keys.

[Claim 7] Processing which performs composite-ization of the above-mentioned

verification pattern based on two or more above-mentioned keys by making information for every key hold as a period classification to the above-mentioned data file. An unauthorized use prevention method of the software for evaluation according to claim 6 performing composite-ization by one processing based on information for every key without performing composite-ization for every key.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the unauthorized use prevention method of the software for evaluation and relates to the unauthorized use prevention method of the software for evaluation which needs use of a data file in detail.

[0002]

[Description of the Prior Art] Generally in circulation of software in order to have just evaluation of software performed widely distributing the evaluation version (software for evaluation) is performed. The evaluation of software of distribution of such an evaluation version is the purpose to the last.

Since it usually lends gratuitously it is necessary for practical use of a routine work etc. to make it not used unjustly.

Generally for prevention of such an unauthorized user restricting [many] a part of functions for example print functions file storing function etc. etc. among the usual functions are made.

[0003] However it is not desirable to restrict storing of data especially for example by the case where storing of the data of a database etc. is important software therefore there are many requests that I want you to remove restriction of a file storing function. To meet such a request as art which moreover aimed at prevention of the unauthorized use how to limit the duration of service and the use count of the software for evaluation conventionally (JPH4-54529A) The method (JPH4-65716A) it becomes impossible to have made it not use improperly when fixed time furthermore passed by judging this password using the password is known.

[0004] Drawing 11 is a figure which is used in the method of limiting the above-mentioned duration of service and use count of the software for evaluation and in which showing an example of the composition of the program file to which prevention from an unauthorized use was performed. In drawing 11 the numerals 1 are main parts for evaluation of application software and 2 is a protection program. The protection program 2 is equipped with the end period information 3 which is information about the end term of evaluation the use count information 4 and the time acquiring means 5. The end period information 3 is calculated and recorded based on time of onset in the time of an evaluation start and is held in a file. The use count information 4 is updated whenever the software for evaluation is

used and it is compared with the number of times of permission which was set up beforehand and recorded. The time acquiring means 5 acquires time of onset when a program starts and usually acquires a hour entry from the clock register of hardware etc. via the time acquisition means of an operating system. And on the basis of such composition the protection program 2 When comparison of warm-up time and the expiration date or a use count judges whether it is in the number of times of permission set up beforehand and has crossed tolerance level at the time of starting of this software for evaluation starting of this software for evaluation is stopped or software files are eliminated.

[0005]

[Problem(s) to be Solved by the Invention] By the way in the case where the method of using the program file shown in drawing 11 is adopted the maker which distributes this software usually carries out about setting out (installation) of the software for evaluation. However in order to get more users to evaluate the software for evaluation by recent years in the distribution by the network or CD-ROM is also made increasingly. If time of onset is set up at the time of installation when a deer is carried out and it distributes by the network or CD-ROM in this way setting out of this time of onset will be left to a user's hand and a possibility that injustice may be performed at the time of setting out of the first end term will arise. For example it will become possible by installing multiple times to use it for a long period of time. Namely the duplicate of a program file is created before setting out of software. It will become possible to use this software continuously by resetting from the program file which saved data after evaluating fixed time or a fixed count and was saved as another evaluation starting evaluation and opening the saved data. Although how to eliminate a data file simultaneously with software is also considered generally a data file will be able to be reproduced and it will become possible to continue and use this software too by reappraising using what was reproduced.

[0006] In order to avoid such inconvenience the method it becomes impossible to have made it not use improperly when fixed time passed by judging this password using the password as mentioned above is also proposed. However it is necessary to set up a term uniformly beforehand not leave setting out of a term to a user in that case so that only the same period may operate. However if a term is set up uniformly in this way for the user who was in acquisition of SOFUTOEA for evaluation the duration of service will become short and will produce the new inconvenience of it becoming impossible to fully evaluate.

[0007] An object of this invention is to provide the unauthorized use prevention method of the software for evaluation which prevented the continuous unauthorized use of the software for evaluation by having been made in light of the above-mentioned circumstances and preventing the continuous unauthorized use of data.

[0008]

[Means for Solving the Problem] In an unauthorized use prevention method of the software for evaluation according to claim 1 in this invention. In an unauthorized

use prevention method of software for evaluation which needs use of a data fileProcessing which adds expiration date information to the above-mentioned data fileand is stored in thisProcessing which reads a day entry in a processor which uses this software for evaluationExpiration date information stored in the above-mentioned data file is compared with a day entry read in the above-mentioned processorWhen a date obtained from this day entry had passed over the expiration date specified using expiration date informationit had processing which restricts use of the above-mentioned data fileand things were made into a solving means of an aforementioned problem.

[0009]According to this inventionexpiration date information is added to a data file which a user createsSince it was made to perform processing which restricts use of the above-mentioned data file when comparing this expiration date information with a day entry read in a processor furthermore and having passed over the expiration dateit becomes possible to prevent a continuous unauthorized use of the above-mentioned data file. Hereas the above-mentioned data filenot only data filessuch as a word processorbut a file which stores setup information of a user at the time of using software is included. Although a large number [creating a data file / software which is not indispensable]this invention can target all the software for evaluation by creating such a file simultaneously with soft starting.

[0010]In an unauthorized use prevention method of the software for evaluation according to claim 5 in this invention. In an unauthorized use prevention method of software for evaluation which needs use of a data fileWhile coding based on a key which depended on the Date of drafting of this data for at least some data stored in the above-mentioned data file and storing this coded data in the above-mentioned data file after thatProcessing which stores a verification pattern which enabled verification of whether to be able to perform composite-ization about this stored data correctly in the above-mentioned data fileProcessing which reads a day entry in a processor which uses this software for evaluation when using data of the above-mentioned data file when using this software for evaluationProcessing which verifies whether composite-ization of the above-mentioned verification pattern is performed based on a key depending on a read day entryand composite-ization about data of the above-mentioned data file can be correctly performed after reading this day entryWhen composite-ization about data of the above-mentioned data file could not be performed correctly and it was verifiedit had processing which restricts use of the above-mentioned data fileand things were made into a solving means of an aforementioned problem.

[0011]While coding and storing at least some data stored in a data file based on a key depending on the Date of drafting of this data according to this inventionA verification pattern which enabled verification of whether to be able to perform composite-ization about this stored data correctly is stored in the above-mentioned data file. And when using data of the above-mentioned data file when using this software for evaluationit is verified whether composite-ization of the above-mentioned verification pattern is performed based on a key depending on a day entry read in a processorand composite-ization about data of the above-

mentioned data file can be performed correctly. When composite-ization about data of the above-mentioned data file could not be performed correctly and it is verifiedprocessing which restricts use of the above-mentioned data file is performed. Thereforewhile becoming possible to prevent a continuous unauthorized use of the above-mentioned data fileit becomes difficult to perform an alteration of this data by coding at least some data.

[0012]That isthere is a possibility that an alteration of expiration date information may be performed by artsuch as reverse engineeringonly by adding expiration date information to a data file simply. In order to avoid such fearhow to encipher expiration date informationfor example is also consideredbut in that casewhen a preservation place is pinpointedthere is a possibility that injustice of reproducing only the portion from a data file of other expiration date etc. may be performed. By carrying out a deerby this inventionas mentioned abovean unauthorized use of SOFUTOEA for evaluation can be more certainly prevented by having made data hard to alter.

[0013]

[Embodiment of the Invention]Hereafterthis invention is explained in detail based on the example of an embodiment. Drawing 1 (a) and (b) is a figure for explaining the example of a 1st embodiment of the unauthorized use prevention method of the software for evaluation according to claim 1 in this inventionand the lineblock diagram of the application file 6 in which drawing 1 (a) serves as software for evaluationand drawing 1 (b) are the lineblock diagrams of the data file 7. The application file 6 consists of the main part 10 of application softwareand the protection program 11and the main part 10 of application software is equipped with the file management means 12. The file management means 12 is a means to store data as a fileand it consists of that with which much software is equippedand a mechanism of the same kind in order to output and input data.

[0014]The protection program 11 is equipped with the time acquisition means 13the data encoding means 14and the data composite-ized means 15. It is for the time acquisition means 13 acquiring the information on the date (time)and is what acquires time of onset when a program starts like the time acquiring means 5 shown in drawing 11A hour entry is acquired from the clock register of hardwareetc. via the time acquisition means of an operating system (processor). The data encoding means 14 changes into the form of a file the data which should be storedandspecificallyadds simply the date evaluated in stored data. The data composite-ized means 15 takes out a day entry from the data in which coding procedure was given. And by having had these time acquisition means 13the data encoding means 14and the data composite-ized means 15 the protection program 11It can be judged whether whether the data in a data file being within a term and a data file are the inaccurate things which passed over the expiration date. The end period information 17 used as the field whichon the other handstores expiration date information other than the user data 16 in the data file 7 as shown in drawing 1 (b) is established.

[0015]Nextwhen it stores data in the user data 16 of the data file 7 using the

application file 6 of such composition and the data file 7 the case where the data file 7 is read further is explained with reference to drawing 2 (a) and (b). In order to store data first judge by the protection program 11 (it is described as Step A1 and following ST-A1) and if it is a new file whether as shown in drawing 2 (a) the data file 7 is a new file. While gaining the date by the time acquisition means 13 what added the permissible hour of use (permission days of consumption) set up beforehand is made into the expiration date and expiration date information is created (ST-A2). If the data file 7 is the existing file the already read expiration date information will be used as it is (ST-A3).

[0016] Next the acquired expiration date information is added to the end period information 17 of the data file 7 and let this be a store file (ST-A4).

Subsequently this is stored in a secondary storage means (graphic display abbreviation) etc. for the obtained store file using the file management means 12 of the main part 10 of application software (ST-A5). For example if the present dates are 95/1/1 (1995 it will be the same as that of the abbreviation on the 1st and the following in January per year) and a permissible hour of use is 30 days when it stores a new file 95/1/31 are stored as expiration date information.

[0017] In order to use the data (data file 7) stored by doing in this way first as shown in drawing 2 (b) the data file 7 is read (ST-B1) and expiration date information is taken out from this data file 7 (ST-B2). When expiration date information is not saved this data file 7 stops that reading as what is used improperly. Next the present date is gained as a day entry by the time acquisition means 13 (ST-B3). And when the date obtained by comparing the obtained date with the expiration date specified using the expiration date information previously acquired from the data file 7 (ST-B4) has passed over the expiration date reading of the above-mentioned data file is stopped.

[0018] If the present dates are 95/2/1 and the expiration date specified using the expiration date information on the data file which it is going to use is 95/1/31 the reading is stopped and it becomes impossible for example to use this data file.

When the obtained date is within the expiration date (i.e. when use of the application file 6 is made within the regular period) reading of the data file 7 is made regularly and evaluation of the application file 6 is presented with it.

[0019] If it is in the unauthorized use prevention method of such software for evaluation Since the expiration date was stored in the data file 7 used by the protection program 11 and it was made to stop reading at the time of reading of this data file 7 when the expiration date had passed The use of the application file 6 which the data file 7 becomes what can use only the fixed time set up beforehand therefore needs use of this data file 7 can also be restricted to fixed time.

[0020] Namely in the conventional method of having stored expiration date information in the program file (application file). By saving the created data file or saving the duplicate of this Even if the expiration date passes and not only a program file (application file) but a data file is eliminated The continuous use of a program file (application file) will become possible as a result by resetting an

evaluation program as new evaluation and opening the reproduced data file. A deer is carried out and since according to the method in this above-mentioned example of an embodiment expiration date information is stored in the data file side even if it reproduces a data file after the expiration date passes use of this data file can be restricted certainly.

[0021] Generally a file may be used for software for many setting out and the configuration file is indispensable to starting of software. Therefore if the method of this invention is applied to this configuration file operation only of the fixed time permitted after the software itself received can be enabled. It can make it possible to also perform evaluation of delivery of the data between two or more software by applying this invention to two or more software.

[0022] Drawing 3 (a) and (b) is a figure for explaining the example of a 2nd embodiment of the unauthorized use prevention method of the software for evaluation according to claim 1 in this invention and the lineblock diagram of the application file 18 in which drawing 3 (a) serves as software for evaluation and drawing 3 (b) are the lineblock diagrams of the data file 19. The place where this example of a 2nd embodiment differs from the above-mentioned example of a 1st embodiment adds expiration date information to a data file faces it storing in this and is at the point which does not only evaluate the date but enciphers this expiration date information. To encryption if it is an operation to a day entry which can be restored the thing of what kind of gestalt may be used and the natural existing encoding technology can be applied.

[0023] The application file 18 used as the software for evaluation shown in drawing 3 (a) It consists of the main part 20 of application software and the protection program 21 like the application file 6 shown in drawing 1 (a) and the main part 20 of application software is equipped with the above-mentioned file management means 12 and the same file management means 22.

[0024] The protection program 21 is equipped with the time acquisition means 23 the data encoding means 24 the data composite-ized means 25 and the expiration date information 26. The time acquisition means 23 functions as the time acquisition means 13 shown in drawing 1 (a) similarly. The data encoding means 24 enciphers expiration date information as mentioned above and it adds it to the data file 19. The data composite-ized means 25 is for returning to the information showing the date which takes out expiration date information from the data file 19 performs the inverse operation of encryption and serves as the original expiration date. The expiration date information 26 is because the expiration date information set up at the time of the first starting is stored. When application file 18 the very thing is distributed the predetermined value which shows un-setting up is set to this expiration date information 26. All over the protection program 21 field on that main memory when this file 18 has started the field (graphic display abbreviation) holding the encoding key Ki to the file opened by execution of this software is provided at this application file 18. The expiration date information 28 used as the field which on the other hand stores expiration date information other than the user data 27 in the data file 19 as shown in drawing 3 (b) is established.

[0025]Nextthe application file 18 of such composition and the data file 19 are usedWhen it stores data in the user data 27 of the data file 19 and reads the data file 19the case where starting of the application file 18 is checked further is explained with reference to drawing 4 (a)(b)and (c). In order to store datafirstjudge by the protection program 21 (ST-C1)and if it is a new filewhether as shown in drawing 4 (a)the data file 19 is a new fileWhile gaining the date by the time acquisition means 23calculate the expiration date by adding the permissible hour of use (permission days of consumption) set up beforehandand This expiration dateA most early thing is made into the regular expiration date among the expiration date (expiration date information) of a file opened by thenand the information based on this expiration date is enciphered by the data encoding means 24 (ST-C2). If the data file 19 is the existing filethe already read expiration date information will be used as it is (ST-C3).

[0026]Nextthe acquired expiration date information is added to the expiration date information 28 on the data file 19and let this be a store file (ST-C4).

Subsequentlythis is stored in a secondary storage means (graphic display abbreviation) etc. for the obtained store file using the file management means 22 of the main part 20 of application software (ST-C5).

[0027]In order to use the data (data file 19) stored by doing in this waylike the case where it is shown in drawing 2 (b)as shown in drawing 4 (b)the data file 19 is read first (ST-D1)and expiration date information is taken out from this data file 19 (ST-D2). When expiration date information is not savedthis data file 19 stops that reading as what is used improperly. Nextthe present date is gained as a day entry by the time acquisition means 23 (ST-D3). And the obtained date and the expiration date specified using the expiration date information previously acquired from the data file 19Namelywhen the date obtained by comparing the expiration date of the origin produced by the inverse operation of encryption being performed by the data composite-ized means 25 (ST-D4) has passed over the expiration dateReading of the above-mentioned data file is stoppedand that use is restrictednamelythe data file 19 and the application file 18 are eliminated in this exampleand these reuses are made impossible (ST-D5). When the obtained date is within the expiration date (i.e.when use of the application file 18 is made within the regular period)like the case where it is shown in drawing 2 (b)reading of the data file 19 is made regularly and evaluation of the application file 18 is presented with it.

[0028]In starting the application file 18as shown in drawing 4 (c)it takes out the expiration date information 26 from the protection program 21 of the application file 18 first as a check of the starting (ST-E1). At this timeit judges whether this expiration date information 26 has been set up (ST-E2)and when not having set upprocessing is advanced to ST-E5 mentioned later. When the expiration date information 26 is set upthe present date is gained as a day entry by the time acquisition means 23 (ST-E3). And when the date obtained by comparing the obtained date with the expiration date set as the above-mentioned expiration date information 26 (ST-E4) has passed over the expiration date. Reading of the

above-mentioned data file is stopped and that use is restricted namely the data file 19 and the application file 18 are eliminated in this example and these reuses are made impossible (ST-E6).

[0029] When the obtained date is within the expiration date (i.e. when use of the application file 18 is made within the regular period) starting of the application file 18 is made regularly. When the expiration date information 26 has not been set up while gaining the date by the time acquisition means 23 the expiration date obtained by calculating the expiration date by adding the permissible hour of use (permission days of consumption) set up beforehand is stored in the expiration date information 26 and the application file 18 is started after ** (ST-E5).

[0030] If it is in the unauthorized use prevention method of such software for evaluation since the data (user data 27) and the program (application file 18) which passed over fixed time do not operate using the expiration date information 28 on the data file 19 use of the software for evaluation over which fixed time passed can be prevented. At this time an unauthorized use can be more certainly prevented by eliminating the application file 18 automatically. Only by adding expiration date information to a data file simply although there is a possibility that the alteration of the expiration date information may be made by various tools in the above-mentioned example of an embodiment. Since it has enciphered to the expiration date information stored in the data file 19 the alteration of expiration date information can be prevented from being made in analyses of a data file such as reverse engineering.

[0031] In order to adopt the most early thing of each expiration date of the data file opened by then as the expiration date of a new data file in the case of the software which can open two or more files simultaneously the trial of duration-of-service extension of the substantial data based on a duplicate within software can be prevented. If a configuration file indispensable to starting of this software is included in a data file at this time it will become effective from this always being read especially. In this example of an embodiment only in the case of the new data file it was made to perform comparison and replacement between the expiration date information on the existing data file but it may be made to perform comparison and replacement between the expiration date information on other existing data files also with the existing data file.

[0032] Drawing 5 (a) and (b) is a figure for explaining the example of a 1st embodiment of the unauthorized use prevention method of the software for evaluation according to claim 5 in this invention and the lineblock diagram of the application file 30 in which drawing 5 (a) serves as software for evaluation and drawing 5 (b) are the lineblock diagrams of the data file 31. The place where this example of an embodiment differs from the example of an embodiment shown in drawing 1 and drawing 3 is a point equivalent to duration of service which changes a key and codes a data file for every date. Here coding is considered as the reversible operation which made the key the parameter to the whole data file and as long as a key is difficult to reason from the result of an operation what kind of thing may be used for it and it may apply existing encoding technology and

compression technology.

[0033]The application file 30 shown in drawing 5 (a) consists of the main part 32 of application software and the protection program 33 and the main part 32 of application software is equipped with the file management means 34. The protection program 33 is equipped with the time acquisition means 35 the key preparing means 36 the data encoding means 37 and the data composite-ized means 38. The time acquisition means 35 is for acquiring the information on the date (time) like the time acquisition means 13 shown in drawing 1 (a). The key preparing means 36 calculates the key for coding based on the day entry acquired by the time acquisition means 35. Hereas a key for coding to the date of fixed time it is the same value and out of a period it forms so that it may differ so that it may mention later. The data encoding means 37 stores the verification pattern which enabled verification of whether to be able to perform composite-ization about this stored data correctly in this data file 31 while it codes the data file which should be stored and stores it in the data file 31. The data composite-ized means 38 is for composite-izing the data in which coding procedure was given by the data encoding means 37 namely is restored to the original gestalt by carrying out the inverse operation of the coded data. On the other hand the user data 39 coded as shown in drawing 5 (b) is stored in the data file 31.

[0034]And by such composition the protection program 33 The verification pattern stored while coding data by the data encoding means 37 by checking whether it is correctly restored by the data composite-ized means 38 It can be judged whether they are whether data is a thing within the expiration date and a file of an unauthorized use. Hereabout the key of the above-mentioned coding as shown in drawing 6 depending on the Date of drafting of data it is changed for every (the example in a figure day by day [30]) fixed time. Thus when in key change it carries out and coding is a period when the same data file also differs in the Date of drafting how coding is carried out also changes. Therefore this data file can restore now correctly only within the same period as the coded day (Date of drafting).

[0035]Next when it stores data in the user data 39 of the data file 31 using the application file 30 of such composition and the data file 31 the case where the data file 31 is read further is explained with reference to drawing 7 (a) and (b). In order to store data first judge by the protection program 33 (ST-F1) and if it is a new file whether as shown in drawing 7 (a) the data file 31 is a new file the time acquisition means 13 -- the date -- gaining (ST-F2) -- an encoding key is created from the gained date (ST-F3). On the other hand in being not a new file but the existing file it uses the already read encoding key as it is (ST-F4).

[0036]Next a verification pattern is added to the data which should be stored and the coding using the above-mentioned encoding key is performed further (ST-F5). Subsequently this is stored in secondary storage (graphic display abbreviation) etc. for the coded data file 31 using the file management means 34 of the application file 30 (ST-F6).

[0037]About the processing which creates the key of coding from the date here. what was created as mentioned above -- fixed time -- it being the somewhat kind

of thing being sufficient as long as future things have the character to differ but. The date is expressed for the numerical value that one day corresponds to 1 for example integer division is done for it in a period (for example days) and how to hash-ize the value etc. can be considered. For example when you store a new file the present dates are 95/1/10 and suppose that the permission expiration date is 30 days. And it will be set to 1108 if what evaluated 95/1/10 is 32882 this will be divided by 30 and integer part will be taken. The key which performs conversion by a hash function etc. to this value once again and becomes $h(1108) = 104325$ is obtained. The Reason for giving a hash function is that change also of the result of coding may become it scarce that a key is a near numerical value and the judgment by a pattern may become difficult.

[0038] In order to use the data which did in this way and was saved at the data file 31 as shown in drawing 7 (b) the data file 31 is read first (ST-G1). Next by the time acquisition means 35 the present date is gained (ST-G2) and a key is defined like the time of coding based on the gained date (ST-G3). Subsequently composite-ization by the data composite-ized means 38 is performed using the defined key and the verification pattern added previously is obtained (ST-G4). Then it judges whether whether what was composite-ized holding the defined verification pattern and the obtained verification pattern are regular (ST-G5) if regular reading processing will be continued and reading processing will be suspended if not regular.

[0039] For example supposing the present dates are 95/2/1 the evaluated thing is set to 33269 and it will be set to 1108 if the integer part of the value produced by dividing this by 30 is taken. If conversion by a hash function etc. is performed to this value $h(1108) = 104325$ will be obtained. Since it becomes the same key as having coded to 95/1/10 as described above therefore composite-ization of data can be performed normally and a verification pattern is also restored normally by this this value can perform next processing normally.

[0040] On the other hand supposing the present dates are 95/2/2 the evaluated thing is set to 33270 and it will be set to 1109 if this is divided by 30 and integer part is taken. The key obtained will be set to $h(1109) = 600455$ if conversion by a hash function etc. is performed to this value. Therefore if composite-ization of data or a verification pattern is tried by making this into a key though natural it cannot composite-ize correctly and of course this cannot be correctly restored about a verification pattern. Therefore it is judged with it being data used for the inaccurate period.

[0041] Since the key of coding by the protection program 33 is changed by the date (Date of drafting) if it is in the unauthorized use prevention method of such software for evaluation it can avoid using the data file coded when it was not a period whose key corresponds. Namely in the conventional method of having duration-of-service information in a program file (application file). By saving the data file created as mentioned above or saving the duplicate of this Although the continuous use of a program file (application file) will become possible as a result according to this method. From the ability of regular composite-ization not to be performed even if it composite-izes based on the day entry which started

this when having passed over the expiration date even if it is reproducing the data file 31. Since it becomes what the data obtained is not regular and cannot be decoded restricting use out of the duration of service of this data file 31 (i.e. the software for evaluation itself) as a result cuts.

[0042] Since there is no possibility that it may be used improperly even if it leaves setting out (installation) of evaluation software to a user according to this method The maintenance service accompanying setting out of a maker etc. etc. are mitigable by exhibiting the software for evaluation generally on a network or distributing it by CD-ROM etc. As mentioned above a file may be used for software for many setting out and to starting of software the configuration file is indispensable. Therefore if this method is applied to this configuration file the software itself can be carried out to the ability only of the fixed time permitted after receiving to be operated. Evaluation of delivery of the data between two or more software can also be performed by applying this method to two or more software. According to this method since data is coded even if it tries to analyze with a data analysis tool etc. it becomes difficult to analyze the contents therefore the unauthorized use of the software for evaluation can be prevented more certainly.

[0043] Drawing 8 (a) and (b) is a figure for explaining the example of a 2nd embodiment of the unauthorized use prevention method of the software for evaluation according to claim 5 in this invention and the lineblock diagram of the application file 40 in which drawing 8 (a) serves as software for evaluation and drawing 8 (b) are the lineblock diagrams of the data file 41. The place where this example of an embodiment differs from the example of an embodiment shown in drawing 5 If evaluation is started to 2/1 when [of that for which the data created by the software for evaluation can use only fixed time now according to the example of an embodiment shown in drawing 5] the one expiration date comes for example by 2/2 Then as it said that it became impossible for the created data to use it only one day when evaluation is started near the date which becomes change of a key the created data is at the point of having improved it becoming impossible to use only in a very short period.

[0044] The application file 40 shown in drawing 8 (a) turns into the application file 30 shown in drawing 5 (a) from the main part 42 of application software and the protection program 43 at the approximately said appearance. On the other hand while the user data 44 in which the data file 41 shown in drawing 8 (b) was coded is stored the classification 45 of the period used as the field which stores the classification of a period is formed. Here coding shall not be performed about the classification 45 of a period. The classification 45 of a period shall assign the classification of A or B for the period divided in this example of an embodiment by turns. This classification can define the digital data of the date from a day entry by whether the number of the values which did integer division in the unit time period is even or the number is odd and a method for example.

[0045] To this application file 40. All over the protection program 43 field on the main memory when this software (application file 40) has started the field (graphic

display abbreviation) holding the encoding key Ki to the file opened by execution of this software is provided. Into the protection program 43 the field (graphic display abbreviation) which stores the present key and the key of the period in front of one of them and each period classification is provided. And it enables it to use it from the start time of evaluation in this example of an embodiment based on such composition by trying composite-ization which used the key of the period in front of one other than the key based on an applicable date in the case of composite-izing of data continuing in two periods. If it does in this way it can evaluate fully within one period set up beforehand at the shortest and can evaluate fully within two periods at the longest. In order to make the trial of the formation of data composite efficient in this example of an embodiment the period was roughly divided into two kinds and the tag for identifying which key is used is provided.

[0046] Since only composite-ization corresponding to the data created within this period T11 in the program P11 which performs reading operation in the period T11 can be performed as shown in drawing 9 (a) if it is in the previous example of a 1st embodiment it is created within this period T11 and it becomes impossible that it to use only data (for example D11). A deer is carried out and in this example of an embodiment since the key of the coding before one is also calculated and it enables it to also perform composite-ization of data based on this key as shown in drawing 9 (b) it becomes available [the data created in the period in front of one]. In the case of the software (application file 40) which can be opened simultaneously two or more files By coding a thing with the shortest expiration date of the keys set as the opened data file 41 as a key for a new file extension of the expiration date by a duplicate within software can also be prevented now.

[0047] Next when it stores data in the user data 44 of the data file 41 using the application file 40 of such composition and the data file 41 the case where the data file 41 is read further is explained with reference to drawing 10 (a) and (b). In order to store data first judge by the protection program 33 (ST-H1) and if it is a new file whether as shown in drawing 10 (a) the data file 31 is a new file While gaining the date by the time acquisition means 35 period classification is calculated from the gained date (ST-H2) and an encoding key is created from the date gained further (ST-H3). However if the data file is opened from the period when the date gained before it is contained in the period in front of one the encoding key and period classification of a period before one will be used. If it is the existing file the encoding key and period classification which were read will be used (ST-H4).

[0048] Next a verification pattern is added to the data which should be stored the coding using the above-mentioned encoding key is performed further and the information on the period classification obtained further is added (ST-H5). Subsequently this is stored in secondary storage (graphic display abbreviation) etc. for the coded data file 41 using the file management means 34 of the application file 40 (ST-H6). For example when it stores a new file the present dates are 95/1/10 the permission expiration date is a maximum of 60 days and the minimum unit of a period may be 30 days. And if what evaluated 95/1/10 is 32882 will divide this by 30 and will take integer part it will be set to 1108. The key which performs

conversion by a hash function etc. to this value once again and becomes it $h(1108) = 104325$ is obtained. Here since the number of 1108 is even period classification is taken as A.

[0049] In order to use the data which did in this way and was saved at the data file 41 as shown in drawing 10 (b) the data file 41 is read first (ST-I1). Next by the time acquisition means 35 the present date is gained and the period classification is defined further (ST-I2). And the defined period classification is compared with the period classification previously stored in the data file 41 (ST-I3) and in being equal it defines an encoding key using the gained present date (ST-I4). In differing the key of the period in front of one and period classification are calculated respectively and let them be a key of composite-izing (ST-I5). Subsequently composite-ization by the data composite-ized means 38 is performed using the defined key and the verification pattern added previously is obtained (ST-I6). Then it judges whether whether what was composite-ized holding the defined verification pattern and the obtained verification pattern are regular (ST-I7) if regular reading processing will be continued and reading processing will be suspended if not regular.

[0050] For example supposing the present dates are 95/2/2 the evaluated thing is set to 33269 and it will be set to 1109 if this is divided by 30 and integer part is taken. Since the number of these values is odd period classification is recognized to be B. When trying to open the file of the period classification A since period classification differs $h(1109-1) = 104325$ which is a key in front of 1 period is used. Then since composite-ization can be normally performed from the ability of the data coded to 95/1/10 to be used therefore a verification pattern is also restored normally it can process normally. Here permission days have become in this example on the 60th. If what was evaluated is set to 3330195/3/5 are divided by 30 and integer part is taken it will be set to 1110 and period classification will be set to A. A key is set to $h(1110) = 775254$ by the file of the period classification A for example. Therefore if composite-ization of the file (period classification is A) of 1/1 creation is tried by making this into a key though natural-izing cannot be carried out [****] correctly and a verification pattern will not be restored correctly either. Therefore it is judged with it being data used for the inaccurate period.

[0051] If it is in the unauthorized use prevention method of such software for evaluation the same effect as the example of an embodiment shown in previous drawing 5 is acquired. Since the data which it is at the evaluation start time and was created is valid till the unit time period of one beyond the use is attained at least about the minimum unit time period therefore duration of service becomes extremely short and time to evaluate can avoid the inconvenience fully secured no longer. In order to adopt a most early thing among each expiration date of the data file opened by then as an encoding key of a new data file The trial of duration-of-service extension of the substantial data based on the duplicate of the data during a file within software (application file 40) can be prevented. If a configuration file indispensable to starting of this software is included in a data file at this time it will

become effective from this always being read especially.

[0052] Although it presupposed that even the data of the unit time period in front of one is effective in this example of an embodiment it may set up permit only the number of arbitrary periods such as before three tracing back to before two.

Although coding is performed to data files other than period classification it may be made to carry out by limiting to some arbitrary data files which do not include period classification. About the protection program 43 as shown in drawing 8 (a) had composition including the data encoding means 37 but. It is also possible to constitute to the unnecessary software of writing so that it may not have this data encoding means 37 in that case the part or all is coded beforehand and data should just be provided.

[0053]

[Effect of the Invention] As explained above the unauthorized use prevention method of the software for evaluation according to claim 1 in this invention Add expiration date information to the data file which a user creates and this expiration date information is further compared with the day entry read in the processor Since it is the method which was made to perform processing which restricts use of the above-mentioned data file when having passed over the expiration date The continuous unauthorized use of the above-mentioned data file can be prevented and thereby the continuous unauthorized use of the software for evaluation with indispensable continuous use of data can be prevented intrinsically.

[0054] The unauthorized use prevention method of the software for evaluation according to claim 5 in this invention While coding and storing at least some data stored in a data file based on the key depending on the Date of drafting of this data The verification pattern which enabled verification of whether to be able to perform composite-ization about this stored data correctly is stored in the above-mentioned data file When using the data of the above-mentioned data file when using this software for evaluation Composite-ization of the above-mentioned verification pattern is performed based on the key depending on the day entry read in the processor Verify whether composite-ization about the data of the above-mentioned data file can be performed correctly and it is made like When composite-ization about the data of the above-mentioned data file could not be performed correctly and it is verified The alteration of this data can be made hard to perform by coding at least some data while being able to prevent the continuous unauthorized use of the above-mentioned data files since it is the method which restricted use of the above-mentioned data file.

[0055] Thus it becomes very effective from it becoming indispensable [a database etc.] especially from having restricted the duration of service of the data file that the data itself continues at a long period of time and it uses the same data as the full-scale use in the applicable field of important software if it is in this invention. When writing the indispensable data of an information set etc. in the application file (software) itself the unauthorized use of software can be prevented by restricting use of this data (impossible). By not only software programs such as a package program but an electronic book MIDI data etc. Supposing fixed reading

software also to what sells data when distributing generally as the advertisement and an object for evaluation this invention can be applied and it becomes possible to make the user before a selling agency purchasing try fixed time and music and information in that case. Not only evaluation but a thing [a thing] it is applicable to a use to limit the trial employment period of data to generally for example it prevents from referring to the correct answer portion in the prize quiz of the software of learning materials or a magazine until a fixed stage comes by this invention can be performed.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is a figure for explaining the example of a 1st embodiment in the invention according to claim 1 and (a) is a lineblock diagram of an application file and (b) is a lineblock diagram of a data file.

[Drawing 2] It is a flow chart for explaining the example of a 1st embodiment in the invention according to claim 1 and the flow chart about the case where (a) stores a data file and (b) are the flow charts about the case where a data file is read.

[Drawing 3] It is a figure for explaining the example of a 2nd embodiment in the invention according to claim 1 and (a) is a lineblock diagram of an application file and (b) is a lineblock diagram of a data file.

[Drawing 4] It is a flow chart for explaining the example of a 2nd embodiment in the invention according to claim 1 The flow chart about the case where the flow chart about the case where (a) stores a data file and (b) read a data file and (c) are the flow charts about the case where starting of an application file is checked.

[Drawing 5] It is a figure for explaining the example of a 1st embodiment in the invention according to claim 5 and (a) is a lineblock diagram of an application file and (b) is a lineblock diagram of a data file.

[Drawing 6] It is an explanatory view about the key of coding.

[Drawing 7] It is a flow chart for explaining the example of a 1st embodiment in the invention according to claim 5 and the flow chart about the case where (a) stores a data file and (b) are the flow charts about the case where a data file is read.

[Drawing 8] It is a figure for explaining the example of a 2nd embodiment in the invention according to claim 5 and (a) is a lineblock diagram of an application file and (b) is a lineblock diagram of a data file.

[Drawing 9] It is a figure showing the relation between a data file and the program under execution and (a) is a related explanatory view in the case of the example of a 1st embodiment in the invention according to claim 5 and a related explanatory view in the case of the example [in / in (b) / the invention according to claim 5] of a 1st embodiment.

[Drawing 10] It is a flow chart for explaining the example of a 2nd embodiment in the invention according to claim 5 and the flow chart about the case where (a) stores a data file and (b) are the flow charts about the case where a data file is

read.

[Drawing 11] It is a lineblock diagram of the conventional application file.

[Description of Notations]

61830 and 40 Application file

71931 and 41 Data file

102032 and 42 Main part of application software

112133 and 43 Protection program

122234 file management means

1323 and 35 Time acquisition means

142437 data encoding means

1525 and 38 Data composite-ized means

16273944 user data

17 End period information

26 and 28 Expiration date information

45 Classification of a period
